



ulm university

universität
uulm

Fakultät für Ingenieurwissenschaften und Informatik
Institut für Datenbanken und Informationssysteme

Bachelorarbeit
im Studiengang Medieninformatik

Usability-Konzept für die Fehlerbehandlung von mobilen Prozess-Aktivitäten

vorgelegt von

Ekaterina Speda

September 2015

1. Gutachter	Prof. Dr. Manfred Reichert
Betreuer:	Rüdiger Pryss
Matrikelnummer	735284
Arbeit vorgelegt am:	25.09.2015

Kurzfassung

Es gewinnt heutzutage immer mehr an Bedeutung, mobile Endgeräte und neue elektronische Kommunikationstechnologien zur Verbesserung der Geschäftsabläufe einzusetzen. Man muss aber damit rechnen, dass die Ausführung der Prozess-Aktivitäten auf Smartphones zu unterschiedlichen Fehlern führen kann. Der Benutzer sollte hierbei über das auftretende Problem und den Lösungsweg rechtzeitig informiert werden.

Im schlimmsten Fall kann ein mobiler User komplett ausfallen, zum Beispiel wegen fehlender Netzwerkverbindung oder bei leerem Akku. Um den Geschäftsvorgang nicht abubrechen, wäre es denkbar, Aktivitäten an andere verfügbare berechnete Nutzer weiterzureichen. Dazu wurde an der Universität Ulm ein umfassendes Konzept zur Delegation entwickelt. Die vorliegende Bachelorarbeit soll dabei zur Unterstützung dienen.

Im Rahmen der Arbeit wurde ein Usability-Konzept entworfen, das die Fehlerbehandlung von mobilen Prozess-Aktivitäten, inklusive die Delegationsprinzipien, widerspiegelt. In Anlehnung an die Design Guidelines der drei Betriebssysteme (Android 5.0, iOS 8 und Windows Phone 8) ergaben sich einige Fehlermeldungen. Usability-Kriterien wie gute Lesbarkeit, Verständlichkeit und Benutzerfreundlichkeit standen bei der Erarbeitung der Benachrichtigungen im Mittelpunkt. Darüber hinaus wurde die Thematik bezüglich Barrierefreiheit angerissen.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sinngemäße Übernahmen aus anderen Werken sind als solche kenntlich gemacht und mit genauer Quellenangabe (auch aus elektronischen Medien) versehen.

Ulm, den 25.09.2015

Ekaterina Speda

Inhaltsverzeichnis

1. Einleitung	1
1.1. Problemstellung	1
1.2. Ziele der Arbeit	2
1.3. Aufbau der Bachelorarbeit	3
2. Anforderungen	5
3. Design Guidelines	7
3.1. Android 5.0	7
3.2. iOS 8	10
3.3. Windows Phone 8	12
4. Entwurf der Fehlermeldungen	17
4.1. Paper Prototyping	17
4.2. Dialogbeschreibung	19
4.2.1. Worklist-View	19
4.2.2. Execution-View	24
4.3. Dialogstrukturdiagramme	29
5. Umfrage	33
5.1. Ziel der Umfrage	33
5.2. Aufbau der Umfrage	33
5.3. Ergebnisse der Umfrage	35
5.3.1. Auswertung	35
5.3.2. Kommentare der Befragten	37
6. Verbesserungspotential	43
6.1. Verbesserungsvorschläge	43
6.2. Verbesserte Mockups	44
7. Fazit	47
A. Anhang	49
A.1. Handskizzen	49
A.1.1. "Low battery"-Fehlermeldung	49
A.1.2. "Lost network connection"-Fehlermeldung	54
A.1.3. "Undefined user location"-Fehlermeldung	59
A.1.4. "Instant shutdown"-Fehlermeldung	65
A.1.5. "Time limit"-Fehlermeldung	70
A.2. Elektronische Mockups	73
A.2.1. Ausgangsdialoge	73
A.2.2. "Lost network connection"-Fehlermeldung	75
A.2.3. "Undefined user location"-Fehlermeldung	79

A.2.4. "Instant shutdown"-Fehlermeldung	83
A.2.5. "Time limit"-Fehlermeldung	88
A.3. Auswertungen der Umfrage	91
A.4. Inhalt der beigelegten CD	122

Abbildungsverzeichnis

1.1. Anwendungsfälle für mobile Delegations- und Backup-Services während der Ausführung, Quelle: [PMR14]	2
3.1. Aufbau einer Fehlermeldung - Android	8
3.2. Ein negatives Beispiel für eine Android-Fehlermeldung	9
3.3. Ein Beispiel für ein iOS Action Sheet mit mehreren Auswahlmöglichkeiten . . .	10
3.4. Windows Phone 8 Dialogfelder, Quelle: [Cora]	14
4.1. Skizze des "Task Selection"-Dialogs und der "Low battery"-Fehlermeldung in der Worklist-View	18
4.2. Skizze des "Task Delegation"-Dialogs und der "Critical low battery"-Fehlermeldung in der Execution-View	18
4.3. Dialog "Task Selection" - Android, iOS, Windows Phone	19
4.4. Dialog "Low battery" während der Aufgabenauswahl - Android, iOS, Windows Phone	20
4.5. Dialog "Task Execution" - Android, iOS, Windows Phone	21
4.6. Dialog "Low battery" während der Aufgabenausführung - Android, iOS, Windows Phone	22
4.7. Dialog "Task Delegation" - Android, iOS, Windows Phone	23
4.8. Dialog "Low battery" während der Aufgabendelegation - Android, iOS, Windows Phone	23
4.9. Bestätigungsdialog "Low battery" während der Aufgabendelegation - Android, iOS, Windows Phone	24
4.10. Dialog "Task Execution" - Android, iOS, Windows Phone	25
4.11. Dialog "Low battery" während der Aufgabenausführung - Android, iOS, Windows Phone	25
4.12. Dialog "Low battery" während der Aufgabenausführung - Android, iOS, Windows Phone	26
4.13. Dialog "Critical low battery" während der Aufgabenausführung - Android, iOS, Windows Phone	27
4.14. Dialog "Task Delegation" - Android, iOS, Windows Phone	27
4.15. Dialog "Low battery" während der Aufgabendelegation - Android, iOS, Windows Phone	28
4.16. Dialog "Critical low battery" während der Aufgabendelegation - Android, iOS, Windows Phone	29
4.17. Dialogstrukturdiagramm "Low battery" - Worklist-View	30
4.18. Dialogstrukturdiagramm "Low battery" - Execution-View	31
5.1. Dialog der Aufgabenauswahl und dazugehörige Fehlermeldung "Low battery" .	34
5.2. Dialog der Aufgabendelegation und dazugehörige Fehlermeldung "Low battery"	34
5.3. Fragen bezüglich der "Low battery"-Fehlermeldung während der Aufgabenauswahl	35
5.4. Auswertung zur allgemeinen Frage über das Alter der Teilnehmer	35
5.5. Auswertung zur allgemeinen Frage über das meist benutzte Betriebssystem . .	36

5.6.	Auswertung zur allgemeinen Frage über das Befolgen der Fehlermeldungen . . .	36
5.7.	Auswertung zur Frage über die Textlänge	37
5.8.	Auswertung zur Frage über die Fortschrittsspeicherung	37
5.9.	Verbesserungsvorschläge zur "Low battery"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	38
5.10.	Verbesserungsvorschläge zur "Low battery"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	39
5.11.	Verbesserungsvorschläge zur "Low battery"-Fehlermeldung während der Aufgabendelegation in der Worklist-View	40
5.12.	Verbesserungsvorschläge zur "Low battery"-Fehlermeldung (2. Stufe) während der Aufgabenausführung in der Execution-View	41
5.13.	Verbesserungsvorschläge zur "Critical low battery"-Fehlermeldung (3. Stufe) während der Aufgabenausführung in der Execution-View	41
5.14.	Verbesserungsvorschläge zur "Low battery"-Fehlermeldung (1. Stufe) während der Aufgabendelegation in der Execution-View	42
5.15.	Verbesserungsvorschläge zur "Critical low battery"-Fehlermeldung (2. Stufe) während der Aufgabendelegation in der Execution-View	42
6.1.	Verbesserungsvorschlag zu der "Low battery"-Fehlermeldung während der Aufgabenausführung in der Ausführungsansicht: Vorher (links) und nachher (rechts) - Android	44
6.2.	Verbesserungsvorschlag zu der "Low battery"-Fehlermeldung während der Aufgabenausführung in der Arbeitslistenansicht: Vorher (links) und nachher (rechts) - Android	44
6.3.	Vorherige "Low battery"-Fehlermeldung während der Aufgabendelegation in der Arbeitslistenansicht - Android	45
6.4.	Verbesserte "Low battery"-Fehlermeldung während der Aufgabendelegation in der Arbeitslistenansicht - Android	45
A.1.	"Low battery"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	49
A.2.	"Low battery"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	50
A.3.	"Low battery"-Fehlermeldung während der Aufgabendelegation in der Worklist-View	51
A.4.	"Low battery"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 1)	52
A.5.	"Low battery"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 2)	52
A.6.	"Critical low battery"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 3)	53
A.7.	"Low battery"-Fehlermeldung während der Aufgabendelegation in der Execution-View	53
A.8.	"Critical low battery"-Fehlermeldung während der Aufgabendelegation in der Execution-View	54
A.9.	"Lost network connection"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	54

A.10."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	55
A.11."Lost network connection"-Fehlermeldung während der Aufgabendelegation in der Worklist-View	56
A.12."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Execution-View	57
A.13.Automatische Delegation: "Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Execution-View	58
A.14."Lost network connection"-Fehlermeldung während der Aufgabendelegation in der Execution-View	59
A.15."Lost network connection"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	59
A.16."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	60
A.17."Lost network connection"-Fehlermeldung während der Aufgabendelegation in der Worklist-View	61
A.18."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Execution-View	62
A.19.Automatische Delegation: "Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Execution-View	63
A.20."Lost network connection"-Fehlermeldung während der Aufgabendelegation in der Execution-View	64
A.21."Instant shutdown"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	65
A.22."Instant shutdown"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	66
A.23."Instant shutdown"-Fehlermeldung während der Aufgabendelegation in der Worklist-View	67
A.24."Instant shutdown"-Fehlermeldung während der Aufgabenausführung in der Execution-View	68
A.25."Instant shutdown"-Fehlermeldung während der Aufgabendelegation in der Execution-View	69
A.26."Time limit"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	70
A.27."Time limit"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 1)	70
A.28."Time limit"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 2)	71
A.29.Automatische Delegation: "Time limit exceeded"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 3)	71
A.30."Time limit"-Fehlermeldung während der Aufgabendelegation in der Execution-View (Stufe 1)	72
A.31.Automatische Delegation: "Time limit exceeded"-Fehlermeldung während der Aufgabendelegation in der Execution-View (Stufe 2)	72
A.32.Dialog der Aufgabenauswahl in der Worklist-View	73
A.33.Dialog der Aufgabenausführung in der Worklist-View	73

A.34.Dialog der Aufgabendelegation in der Worklist-View	74
A.35.Dialog der Aufgabenausführung in der Execution-View	74
A.36.Dialog der Aufgabendelegation in der Execution-View	75
A.37."Lost network connection"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	75
A.38."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	76
A.39."Lost network connection"-Fehlermeldung während der Aufgabendelegation in der Worklist-View (Teil 1)	76
A.40."Lost network connection"-Fehlermeldung während der Aufgabendelegation in der Worklist-View (Teil 2)	77
A.41."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 1)	77
A.42."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 2, Teil 1)	78
A.43."Lost network connection"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 2, Teil 2)	78
A.44."Lost network connection"-Fehlermeldung während der Aufgabendelegation in der Execution-View	79
A.45."Undefined user location"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	79
A.46."Undefined user location"-Fehlermeldung während der Aufgabenausführung in der Worklist-View	80
A.47."Undefined user location"-Fehlermeldung während der Aufgabendelegation in der Worklist-View (Teil 1)	80
A.48."Undefined user location"-Fehlermeldung während der Aufgabendelegation in der Worklist-View (Teil 2)	81
A.49."Undefined user location"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Teil 1)	81
A.50."Undefined user location"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Teil 2)	82
A.51."Undefined user location"-Fehlermeldung während der Aufgabendelegation in der Execution-View	82
A.52."Instant shutdown"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View (Teil 1)	83
A.53."Instant shutdown"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View (Teil 2)	83
A.54.Android-Systemmeldung nach der Anzeige der "Instant shutdown"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View (Teil 3)	84
A.55."Instant shutdown"-Fehlermeldung während der Aufgabenausführung in der Worklist-View (Teil 1)	84
A.56.Android-Systemmeldung nach der Anzeige der "Instant shutdown"-Fehlermeldung während der Aufgabenausführung in der Worklist-View (Teil 2)	85
A.57."Instant shutdown"-Fehlermeldung während der Aufgabendelegation in der Worklist-View (Teil 1)	85

A.58."Instant shutdown"-Fehlermeldung während der Aufgabendelegation in der Worklist-View (Teil 2)	86
A.59.Android-Systemmeldung nach der Anzeige der "Instant shutdown"-Fehlermeldung während der Aufgabendelegation in der Worklist-View (Teil 3)	86
A.60."Instant shutdown"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Teil 1)	87
A.61.Android-Systemmeldung nach der Anzeige der "Instant shutdown"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Teil 2)	87
A.62."Time limit"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View	88
A.63."Time limit"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 1)	88
A.64."Time limit"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 2)	89
A.65.Automatische Delegation: "Time limit exceeded"-Fehlermeldung während der Aufgabenausführung in der Execution-View (Stufe 3)	89
A.66."Time limit"-Fehlermeldung während der Aufgabendelegation in der Execution-View (Stufe 1)	90
A.67.Automatische Delegation: "Time limit exceeded"-Fehlermeldung während der Aufgabendelegation in der Execution-View (Stufe 2)	90
A.68.Auswertung der Umfrage 1	92
A.69.Auswertung der Umfrage 2	93
A.70.Auswertung der Umfrage 3	94
A.71.Auswertung der Umfrage 4	95
A.72.Auswertung der Umfrage 5	96
A.73.Auswertung der Umfrage 6	97
A.74.Auswertung der Umfrage 7	98
A.75.Auswertung der Umfrage 8	99
A.76.Auswertung der Umfrage 9	100
A.77.Auswertung der Umfrage 10	102
A.78.Auswertung der Umfrage 11	103
A.79.Auswertung der Umfrage 12	104
A.80.Auswertung der Umfrage 13	105
A.81.Auswertung der Umfrage 14	106
A.82.Auswertung der Umfrage 15	107
A.83.Auswertung der Umfrage 16	108
A.84.Auswertung der Umfrage 17	109
A.85.Auswertung der Umfrage 18	110
A.86.Auswertung der Umfrage 19	112
A.87.Auswertung der Umfrage 20	113
A.88.Auswertung der Umfrage 21	114
A.89.Auswertung der Umfrage 22	115
A.90.Auswertung der Umfrage 23	116
A.91.Auswertung der Umfrage 24	117
A.92.Auswertung der Umfrage 25	118
A.93.Auswertung der Umfrage 26	119
A.94.Auswertung der Umfrage 27	120

A.95.Auswertung der Umfrage 28	121
--	-----

Zu einem guten Ende gehört auch ein guter Beginn.

Konfuzius, (551 - 479 v. Chr.)

1

Einleitung

In diesem Kapitel werden Problemstellung, Zielsetzung und Aufbau der vorliegenden Bachelorarbeit besprochen.

1.1. Problemstellung

Die Abwicklung der täglichen Geschäftsroutinen kann mithilfe von mobilen Anwendungen unterstützt werden. Während der Ausführung von Prozess-Aktivitäten auf Smart-Mobilgeräten ist das Auftreten verschiedener Fehler möglich. Es handelt sich hierbei um benutzer- und umgebungsbezogene Probleme, wie etwa eine nicht definierte Benutzerlokalisierung oder eine schlechte Netzwerkverbindung. Damit der User notwendige Gegenmaßnahmen schnellstmöglich durchführen kann, müssen diese "Schwachstellen" entsprechend behandelt und in Form von Fehlermeldungen auf dem Bildschirm ausgegeben werden. Die Darstellung der Benachrichtigungen hängt außerdem von unterschiedlichen prozessbezogenen Faktoren ab, wie zum Beispiel der Zeitbeschränkung.

Die Tabelle 1.1 listet die für diese Arbeit relevanten Probleme auf.

Benutzerbezogen	Umgebungsbezogen	Prozessbezogen
Sofortiges Ausschalten	Schlechte Verbindung	Zeitbeschränkung
Nicht definierte Benutzerlokalisierung	Akku schwach	

Tabelle 1.1.: Probleme bei der Ausführung mobiler Anwendungen, Quelle: [PMR14]

Eine Lösung für die aufgeführten Probleme besteht darin, Aktivitäten an andere angemeldete Verantwortliche zu delegieren, falls ein mobiler Benutzer ausfällt. Dazu wurde an der Universität Ulm ein umfangreiches Konzept zur Weiterreichung ausgearbeitet [PMR14] [SSP⁺14a] [PTR10] [PTKR10] [PMR13]. Um die Robustheit und Flexibilität bei der Aufgabenausführung

auf Smartphones zu garantieren, sahen die Entwickler einen automatischen Delegations-Service vor. Des Weiteren musste ein Backup-Service das Sichern und Wiederherstellen der Daten übernehmen, wenn kein für die Aufgabe zuständiger User verfügbar ist. Die beiden Ansätze sollten unter anderem in Fehlermeldungen nachvollziehbar veranschaulicht werden.

Das Diagramm unten zeigt die Anwendungsfälle für mobile Delegations- und Backup-Services während der Ausführung.

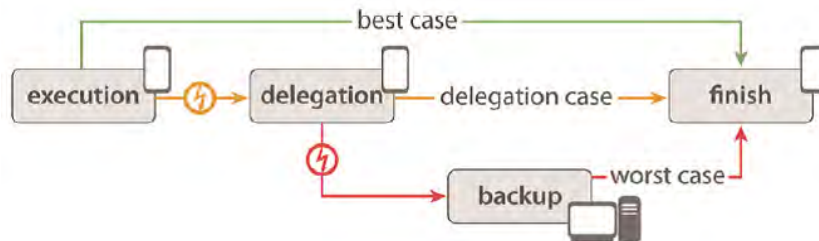


Abbildung 1.1.: Anwendungsfälle für mobile Delegations- und Backup-Services während der Ausführung, Quelle: [PMR14]

Zur Zeit befassen sich mehrere wissenschaftliche Arbeiten mit dem aufwendigen Entwurf einer Informationsarchitektur, die die oben beschriebene Delegation sowie die Datensicherung zuverlässig integriert [PMLR15] [SPSR15] [SPR15] [GSP⁺14] [SSP⁺14b] [CNB⁺13] [GPSR13] [IRLP⁺13] [RLPL⁺13] [PLRH12] [RPR11]. Denn mit einer gut durchdachten dienstorientierten Informationsorganisation ist es möglich, eine hohe Akzeptanz unter den Anwendern zu erreichen.

1.2. Ziele der Arbeit

Ziel der nachfolgenden Bachelorarbeit ist es, ein Usability-Konzept zu erarbeiten, das die Fehlerbehandlung von mobilen Prozess-Aktivitäten für Endanwender ansprechend präsentiert. Die Delegations- und Backup-Services sollen dabei in benutzerfreundlicher Weise visualisiert werden. Als Ergebnis sind digitale Mockups zu erwarten, die in Anlehnung an die Design Guidelines der drei Betriebssysteme (Android 5.0, iOS 8 und Windows Phone 8) gestaltet sind.

Der Entwurf der Benachrichtigungen muss den Anforderungen in Bezug auf Aussehen, Bedienung und Verständlichkeit entsprechen und auf die Erwartungen der mobilen Benutzer abgestimmt sein. Dazu ist am Ende der Arbeit eine Umfrage geplant. Diese soll herausfinden, inwieweit die Teilnehmer mit der Optik und den Formulierungen der entstandenen Fehlermeldungen zufrieden sind.

Es ist von wesentlicher Bedeutung, das bestehende Verbesserungspotenzial frühzeitig zu erkennen. Deshalb bekommen die Befragten die Möglichkeit, eigene Optimierungsvorschläge zu hinterlassen. Die denkbaren Veränderungen sollen extra ausdiskutiert werden.

1.3. Aufbau der Bachelorarbeit

Nach dieser Übersicht über die Struktur des entwickelten Usability-Konzepts ist die Arbeit in sechs weitere Kapitel aufgeteilt. Das Kapitel 2 erläutert die Anforderungen an die Darstellung der Fehlerbehandlung von mobilen Prozess-Aktivitäten. Es geht hierbei um nichtfunktionale Anforderungen. Im dritten Abschnitt auf den Seiten 7 bis 15 werden die Gestaltungsrichtlinien von Android 5.0, iOS 8 und Windows Phone 8 vorgestellt. Darauf basiert der Entwurf der Fehlermeldungen für die drei erwähnten Betriebssysteme. Die Handskizzen, die elektronischen Mockups und die Dialogbeschreibungen folgen im Kapitel 4. Anschließend bildet ein Dialogstrukturdiagramm die Dialogabläufe ab. Das Kapitel 5 befasst sich mit einer Online-Umfrage, deren Auswertung im Kapitel 6 berücksichtigt wird. Dieses ergänzt das Usability-Konzept mit einigen Verbesserungsvorschlägen für die künftige Fehlerbehandlung. Das letzte Kapitel 7 rundet die Bachelorarbeit mit einer Zusammenfassung ab.

2

Anforderungen

Das folgende Kapitel beinhaltet nichtfunktionale Anforderungen an die Darstellung der mobilen Fehlermeldungen. Diese sind im Hinblick auf Aussehen, Verständlichkeit und Sicherheit wichtig. Zusätzlich dienen sie dazu, die Akzeptanz des Systems gegenüber dem Anwender zu erhöhen.

Die Anforderungen werden in Tabelle 2.1 zusammengestellt.

Nr.	Anforderung	Beschreibung
1	Einheitliches Aussehen der Benachrichtigungen	Die Fehlermeldungen eines Betriebssystems sollen ein einheitliches Design aufweisen.
2	Erkennbarkeit	Die Fehlermeldungen aller drei Betriebssysteme sollen vom Benutzer als solche sofort identifiziert werden.
3	Einhaltung der Design Guidelines	Um ein einheitliches Erscheinungsbild zu erreichen, sollen die Gestaltungsrichtlinien der drei Betriebssysteme (Android 5.0, iOS 8 und Windows Phone 8) eingehalten werden.
4	Optimales Layout	Die Meldungen sollen aufgrund ihrer Schriftgröße, Wort- und Zeilenabstände leicht lesbar sein.
5	Optimale Textlänge	Eine Fehlermeldung sollte möglichst kurz gehalten werden, damit der Nutzer sie zu Ende liest. Trotzdem soll sie aussagekräftig bleiben.
6	Verständlichkeit der Formulierungen	Die Texte der Fehlermeldungen sollen verständlich formuliert sein. Insbesondere sind Fachbegriffe zu vermeiden.
7	Hervorhebung	Die Fehlermeldungen sollen sich gut vom Hintergrund abheben.
8	Eindeutige Benennung der Buttons	Es ist auf eine eindeutige Benennung der Buttons zu achten, um keine Verwirrung unter den Usern auszulösen.
9	Verständlichkeit der Handlung	Die durchzuführende Handlung muss durch Schaltflächenbezeichnungen klar wiedergegeben werden. Der Nutzer soll verstehen, wohin der Klick des Buttons führt. Die Buttons mit unterschiedlicher Funktionalität dürfen nicht gleich bezeichnet werden.

10	Pragmatische Hilfestellung	Die Fehlermeldungen sollen beim Lösen des Problems hilfreich sein.
11	Benutzerfreundlichkeit	Es ist ein benutzerfreundlicher Aufbau der Formulierungen zu erwarten. Der Nutzer soll in höflicher Form angesprochen werden.
12	Zuverlässigkeit	Das Delegations-Konzept soll in den Fehlermeldungen nachvollziehbar abgebildet sein. Der User soll die Möglichkeit haben, das System wieder in einen korrekten Zustand zu überführen bzw. es muss der Übergang in einen gesicherten Zustand gewährleistet werden.
13	Sicherheit	Die Fehlermeldungen sollen das Backup-Konzept berücksichtigen. Der Nutzer muss wissen, dass die eingegebenen Informationen im Fehlerfall zwischengespeichert werden.
14	Unterscheidung nach Dringlichkeit	Der User soll die kritischen Fehler sofort als solche erkennen, damit er rechtzeitig reagieren kann.

Tabelle 2.1.: Tabelle nichtfunktionaler Anforderungen

3

Design Guidelines

Was die Fehlermeldungen betrifft, sind diese nur dann sinnvoll, wenn sie strukturiert und einheitlich umgesetzt sind. Eine Fehlermeldung soll vom User als solche sofort erkannt werden und beim Lösen des Problems helfen. Eine freundliche, benutzerorientierte Gestaltung ist ebenso von großer Bedeutung.

Jedes mobile Betriebssystem hat seine Besonderheiten. Die Design Guidelines dienen dabei als anerkannter Leitfaden, um eine verbesserte Benutzerführung sowie eine uniforme Optik zu erreichen. Das Einhalten der vordefinierten Gestaltungsrichtlinien fördert die Benutzbarkeit und könnte deshalb entscheidend sein, ob der Anwender eine Warnung zu Ende liest.

Aus diesem Grund werden im aktuellen Kapitel die Design Guidelines der drei Plattformen, der Android 5.0, iOS 8 und Windows Phone 8, vorgestellt. Die allgemeinen Regeln für Verwendung der Meldungen und deren Aufbau stehen hierbei im Mittelpunkt. Weiterhin wird ein spezielles Augenmerk auf die Barrierefreiheit gelegt.

Im Abschnitt 3.1 sind die relevanten Empfehlungen für das Android Betriebssystem zu lesen. Der zweite Teil beinhaltet die zum Thema passenden iOS Guidelines (siehe Seite 10). Im abschließenden Unterkapitel 3.3 folgt eine Beschreibung der entsprechenden Gestaltungsrichtlinien für die Windows Phone Plattform.

3.1. Android 5.0

Verwendung der Meldungen

Die Entwickler der Google-Richtlinien für Android unterstreichen die Wichtigkeit der informativen Rolle der Warnungen: "Alerts inform the user about a situation or action that requires their confirmation or acknowledgement before proceeding." [Ince] Man sollte die Fehlermeldungen mit Bedacht verwenden, weil sie den Programmablauf unterbrechen: "Not every choice, setting, or detail warrants interruption and prominence." [Ince] Zum Vermeiden der überflüssigen Mitteilungen werden zusätzliche Menüpunkte bzw. erweiterte Optionen für ergänzende Informationen im Inhaltsbereich der Applikation empfohlen.

Eine Meldung muss im Fokus des Users sein, daher dürfen andere Elemente sie nicht überdecken. Auf die verschachtelten Fenster und scrollbaren Bereiche sollte nach Möglichkeit verzichtet werden: "Instead, consider alternate containers or layouts that are optimized for reading or interacting with significant amounts of content." [Ince] Insbesondere wird davon abgeraten, die Aktionen einzuführen, die von der Applikation wegnavigieren und die gegenwärtige Aufgabe in einen nichtdeterminierten Zustand bringt. Ein Beispiel dazu wäre ein "Learn more"-Button, der den Zugang zu der Hilfedokumentation erlaubt. Eine Lösung dazu ist: "If a little more detail or

explanation is needed for the dialog contents, use in-line expansion within the dialog. If more extensive information or explanation is needed for the dialog content, provide that information prior to entering the dialog." [Ince]

Aufbau einer Meldung

Eine Meldung lässt sich grob in die drei Bereiche einteilen: Titel, Inhalt und Aktionen (siehe Abb. 3.1). Die Design Guidelines geben eine Richtung vor:

- Clearly state a potential result in the dialog title.
- The dialog content should directly relate to the title or choices.
- Present clear choices. [Ince]

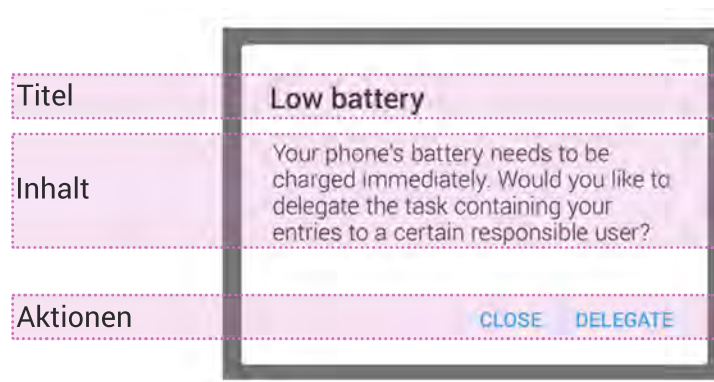


Abbildung 3.1.: Aufbau einer Fehlermeldung - Android

Eine Empfehlung der Herausgeber der Android Design Guidelines ist, einen Titel sparsam und nur in Situationen mit hohem Risiko, beispielsweise wegen bevorstehendem Datenverlust, einzusetzen. Wenn eine Überschrift unvermeidbar ist, dann sollte sie am besten als eine klare Frage formuliert sein. Alternativ eignet sich dafür eine aussagekräftige Feststellung mit einer prägnanten Erklärung im Inhaltsbereich. "A user should be able to skip the content completely and still have a clear idea of what choices are available based on the title and the text of the action buttons," [Ince] klären die Autoren der Guidelines auf. Sie raten nachdrücklich davon ab, Entschuldigungen, mehrdeutige Feststellungen oder missverständliche Fragen im Titelbereich zu benutzen: "For example, don't use 'Warning!' or 'Are you sure?'" [Ince]

Wie bereits erwähnt, befindet sich typischerweise im Inhaltsbereich einer Meldung eine textuelle Problembeschreibung. Oft aber werden hier auch die Steuerelemente der Benutzeroberfläche platziert.

Die Auswirkungen des aufgetretenen Problems müssen für den Benutzer klar sein, genauso wie die vorzunehmenden Handlungen. Sie werden im Aktionsbereich durchgeführt. In erster Linie sollen die Aktionen dem User helfen, eine Entscheidung auf möglichst einfache Weise zu treffen. Für die Benennung der Buttons eignen sich daher klare, eindeutige Verben bzw. Verbalphrasen, die im direkten Zusammenhang mit dem Titel und dem Inhalt der Fehlermeldung stehen.

Die Android-Entwickler unterscheiden generell zwischen bejahenden und ablehnenden Aktionen. Dies könnten etwa eine Lösungsbestätigung oder ein Verwerfen gewisser Einstellungen sein. Bejahende Aktionen setzen den Prozess fort und sollen auf der rechten Seite der Android-

Fehlermeldung zu finden sein. Ablehnende Aktionen führen den Nutzer zum Ursprungsbildschirm oder zu dem vorherigen Schritt zurück. Die Buttons für die Ablehnung werden links von den bejahenden Schaltflächen angebracht. Des Weiteren weisen die Richtlinien darauf hin, dass die Buttons, hinter denen eine ablehnende Handlung steht, nie im deaktivierten Zustand erscheinen sollten: "Affirmative actions are more likely to be disabled until a choice is made. Dismissive actions are never disabled." [Ince]

Der Meinung der Android-Designer nach, sollte man sich nur in den seltensten Fällen für Warnungen mit nur einem Button entscheiden. Sie bieten dem User keine Auswahl, unterbrechen aber den Ablauf. Es sollten jedoch auch nicht zu viele Buttons angeboten werden. Es gilt die Faustregel: Nicht mehr als zwei Aktionen pro Meldung.

Barrierefreiheit

Eine barrierefreie Nutzung ist nicht nur für Menschen mit Behinderung von Vorteil. Sie trägt auch zu einer besseren Benutzerfreundlichkeit für alle Endanwender bei. Die Design Guidelines der Android Plattform schreiben die genaue Fenstergröße, den Abstand zwischen den Bereichen und die Größe der Buttons vor, um die Meldungen hindernisfrei zu gestalten: "To ensure usability for people with disabilities, make sure that your buttons have a minimum height of 36dp, but that the touchable target has a minimum height of 48dp." [Ince]

Die vom System vorgegebenen Farben können in die Anwendungsdesignfarben überschrieben werden. Dabei ist es jedoch wichtig zu beachten, dass die Kombination der Akzentfarben keine unbeabsichtigte Bedeutung vermittelt. Alle Texte müssen außerdem ausreichende Kontrastverhältnisse haben, um die Bestimmungen der Richtlinien für Barrierefreiheit zu erfüllen.

So warnen im Beispiel unten (siehe Abb. 3.2) die roten Buttons vor Gefahren, obwohl deren Betätigung zu keiner Zerstörung führen. Der mangelhafter Kontrast zwischen Text und Hintergrundfarbe im Inhaltsbereich schränkt die Lesbarkeit erheblich ein.

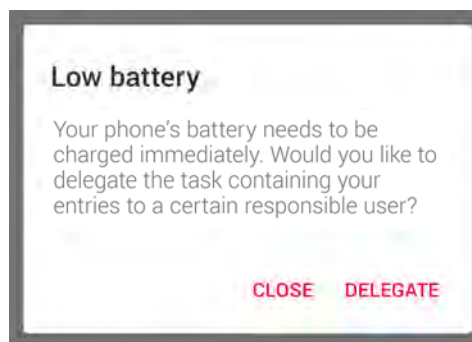


Abbildung 3.2.: Ein negatives Beispiel für eine Android-Fehlermeldung

In den Accessibility Guidelines für Android Designer sind Kontrastverhältnisse festgelegt: "Use a contrast ratio of 4.5:1 between the background and text or critical elements to allow users to read text more easily." [Incd] Text mit geringer Schriftgröße braucht viel Kontrast, im Gegenteil dazu dürfen die großen Überschriften und deren Hintergründe in einer breiten Palette von Farben dargestellt werden. Weitere Vorgaben zu der Navigation, Lesbarkeit sowie Anleitungen und Feedback sind online unter "Accessibility" [Incd] nachzulesen.

3.2. iOS 8

Verwendung der Meldungen

Die Hauptaufgabe einer Warnung wird von den iOS Human Interface Guidelines folgendermaßen definiert: "An alert gives people important information that affects their use of an app or the device." [Incc] Ähnlich wie die Android-Designer, empfehlen die Entwerfer des iOS Betriebssystems unnötige Fehlermeldungen zu vermeiden. Sie sagen zudem, dass je seltener eine Benachrichtigung auf dem Bildschirm erscheint, umso ernsthafter wird sie vom User wahrgenommen. Deshalb ist es ratsam, die Meldungen nach deren Seriosität auszusortieren und nur die mit den kritischsten Informationen und den wichtigsten Entscheidungsvorschlägen zu verwenden.

In den iOS Richtlinien werden außerdem einige Szenarien für eine positive Umwandlung sinnloser Warnungen beschrieben. Zum Beispiel, wenn eine Auskunft über die Standardfunktionen der Anwendung anzuzeigen ist, sollte man die Informationen auf eine auffallende und auf den Stil der App abgestimmte Weise innerhalb der Benutzeroberfläche designen, ohne dass ein Benachrichtigungsfenster extra eingeblendet wird. Oder anstatt den User nach einer Aufgabenbestätigung zu fragen, wäre es besser, ihm ein Action Sheet mit einer Reihe von Auswahlmöglichkeiten anzubieten (siehe Abb. 3.3).

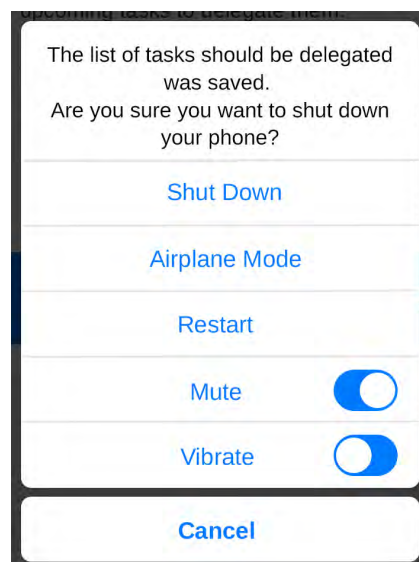


Abbildung 3.3.: Ein Beispiel für ein iOS Action Sheet mit mehreren Auswahlmöglichkeiten

Aufbau einer Meldung

Der Aufbau einer iOS-Fehlermeldung ähnelt dem einer Android-Fehlermeldung (siehe Abb. 3.1), mit dem Unterschied, dass hier ein Titel erforderlich ist, die Nachricht im Inhaltsbereich aber ausgelassen werden darf. Darüber hinaus soll ein Titel es unnötig machen, eine Nachricht hinzuzufügen.

Die Überschrift könnte, zum Beispiel, als eine Frage oder als ein bestenfalls einzeiliges Satzfragment formuliert sein: "A short, informative statement tends to be easier to understand than a complete sentence." [Incc] Die Verfasser der iOS Design Guidelines merken an, dass ein langer

Titel das schnelle Lesen erschwert. Er darf jedoch nicht aus einem Einzelwort bestehen, da solche Bezeichnungen wie "Fehler" oder "Warnung" selten hilfreich sind.

Ein freundlicher Umgangston der Meldungen fördert die Benutzerfreundlichkeit der Applikation. Die Texte sollen nicht anklagend oder beurteilend klingen, wenn negative Nachrichten geliefert werden müssen. Aber "[...] it's better to be negative and direct than it is to be positive but oblique." [Incc]

Beim Entwerfen einer iOS-Fehlermeldung sollte man die Rolle der Groß- und Kleinschreibung eines Textes nicht unterschätzen. Es ist zwischen titelartiger und satzartiger Großschreibung zu differenzieren:

- Titel capitalization means that every word is capitalized, except articles, coordinating conjunctions, and prepositions of four or fewer letters when they aren't the first word.
- Sentence-style capitalization means that the first word is capitalized, and the rest of the words are lowercase unless they are proper nouns or proper adjectives. [Incc]

Die korrekte Interpunktion ist ebenso wichtig. Die Tabelle 3.1 zeigt die iOS Guidelines zur Groß- und Kleinschreibung sowie Verwendung der Satzzeichen.

Wenn der Titel der Meldung...	Verwende...
ein Satzteil oder ein einzelner Satz, jedoch kein Fragesatz, ist	titelartige Großschreibung und kein Satzzeichen
ein Fragesatz ist	satzartige Großschreibung und ein abschließendes Fragezeichen
aus zwei oder mehr Sätzen besteht	satzartige Großschreibung und passende Zeichensetzung am Ende jedes Satzes

Tabelle 3.1.: iOS Empfehlungen zur Groß- und Kleinschreibung sowie Interpunktion, Quelle: [Incc]

Entscheidet man sich dennoch für eine Nachricht im Inhaltsbereich, ist diese aus vollständigen kurzen Sätzen zu bilden. Eine oder zwei Textzeilen reichen aus: "If the message is too long, it will scroll, giving users a poor experience." [Incc] Für die Schreibweise wird eine satzartige Großschreibung und passende Zeichensetzung am Satzende empfohlen.

Die iOS Entwickler raten davon ab, im Inhaltsbereich zu erwähnen, auf welchen Button zu klicken ist. Der User kann die Schlussfolgerungen darüber aus einer eindeutigen Problembeschreibung und einer logischen Bezeichnung der Buttons ziehen. Eine Schaltflächenbeschriftung besteht gemäß den Richtlinien aus einem Verb bzw. einer Zweiwort-Verbphrase. Diese ist ohne Anführungszeichen dargestellt und titelartig großgeschrieben. Die weiteren Vorgaben sind: "Use 'OK' for a simple acceptance option if there is no better alternative. Avoid using 'Yes' or 'No'." [Incc]

Was die Anzahl der Buttons pro Fehlermeldung betrifft, stimmen die Meinungen der iOS und Android Designer überein: Eine Meldung mit einem einzigen Button ist weniger nützlich, weil sie die Endanwender informiert, ohne ihnen jegliche Kontrolle über die Situation zu geben.

Zwei Buttons pro Meldung sind optimal, weil es für Benutzer am leichtesten ist, zwischen zwei Alternativen zu wählen. "An alert that contains three or more buttons is significantly more complex than a two-button alert and should be avoided as much as possible," [Incc] geben die Autoren der iOS Design Guidelines eine ausführliche Erklärung, "If you add too many buttons to an alert, it can cause the alert to scroll, which is a bad user experience." [Incc]

Hinter dem Konzept für die Platzierung der iOS Schaltflächen steht eine andere Idee als im Kapitel 3.1. Man ermittelt zunächst den bevorzugten Button, den der User am wahrscheinlichsten anklicken wird. Er führt die meist gewünschte Aktion aus und verursacht bei der versehentlichen Betätigung die wenigsten Probleme. Wenn der bevorzugte Button eine nicht-destruktive Handlung durchführt, soll er auf der rechten Seite des Dialogs mit zwei Schaltflächen stehen. Der Button, der diese Aktion abbricht, sollte links sein. Dagegen ist für den bevorzugten Button mit einer zerstörerischen Wirkung der linke Platz in der Meldung vorgeschrieben. Der annullierende Button sollte dabei rechts angebracht werden.

Die weiteren Details zur empfohlenen Gestaltung der Fehlermeldungen sind online [Incc] verfügbar.

Barrierefreiheit

Das iOS Betriebssystem liefert einige eingebaute Funktionen, die Nutzern mit Hör- und Sehbehinderung ein erleichtertes Bedienen ermöglichen. Dies sind VoiceOver, Guided Access, Text to Speech uvm. Jeder Anwender kann daraus einen praktischen Nutzen ziehen.

Die iOS-Fehlermeldungen sind standardmäßig barrierefrei. Das betrifft vor allem die drei im vorigen Kapitel genannten Elemente: Titel der Meldung, Nachricht im Inhaltsbereich und Bezeichnungen der Buttons. Sie werden vor allem vom Bildschirmreader unterstützt:

If VoiceOver is activated, it speaks the word "alert" when an alert is shown, then speaks its title followed by its message if set. As the user taps a button, VoiceOver speaks its title and the word "button." As the user taps a text field, VoiceOver speaks its value and "text field" or "secure text field." [Incb]

Weitere Informationen zur barrierefreien iOS Gestaltung findet man im Accessibility Programming Guide for iOS [Inca].

3.3. Windows Phone 8

Verwendung der Meldungen

Möchte ein Designer eine effektive Windows Meldung kreieren, sollte er darauf achten, dass die Nachricht den Anwender über das auftretende Problem informiert, die Ursache des Fehlers erwähnt und eine Lösung anbietet. Als Folge davon muss der Nutzer entweder eine bestimmte Aktion durchführen oder sein Verhalten ändern. Die Autoren der Microsoft Guidelines betonen die ausschlaggebende Rolle der gut geschriebenen Fehlermeldungen für die User Experience:

Well-written, helpful error messages are crucial to a quality user experience. Poorly written error messages result in low product satisfaction, and are a leading cause of avoidable technical support costs. Unnecessary error messages break users' flow. [Corb]

Eine gut geschriebene Windows Fehlermeldung ist [Corb]:

- Relevant: Die Meldung präsentiert ein Problem, das der Nutzer lösen will.
- Benutzerorientiert: Das Problem wird aus der Sicht des Benutzers, nicht des Programmierers erklärt.
- Kurz: Die Meldung ist so kurz wie möglich, aber nicht kürzer.
- Klar: Die Meldung ist in einfacher Sprache verfasst, so dass der Benutzer das Problem und seine Lösung leicht verstehen kann.
- Präzise: Die Meldung beschreibt das Problem in präziser Sprache, mit der genauen Angabe von Namen, Positionen und Werten der beteiligten Objekte.
- Höflich: Der Benutzer darf sich nicht unwohl oder dumm fühlen.
- Selten: Die Fehlermeldung wird selten angezeigt. Oft gezeigte Fehlermeldungen sind ein Zeichen schlechten Designs.

Microsoft Windows ist streng in Vorschriften: "Often the best error message is no error message." [Corb] Es ist es nicht ratsam, Fehler anzuzeigen, die der Benutzer als akzeptabel betrachtet, oder innerhalb einer Fehlermeldung einen komplizierten Fehlerbehebungsprozess zu erklären. Viele Details oder dramatische Formulierungen können auf den Nutzer eine negative Wirkung haben.

Damit unnötige Fehlermeldungen vermieden werden können, schlagen Designer vor, eine Reihe von Fragen vor dem Entwicklungsbeginn zu beantworten und erst danach zu reagieren (siehe Tabelle 3.2).

Frage	Antwort	Lösungsvorschlag
Ist das Problem bereits aufgetreten?	Nein	Es liegt kein Fehler vor. Falls es sich um ein mögliches zukünftiges Problem handelt, soll eine Warnung verwendet werden.
Kann das Problem verhindert werden, ohne einen Irrtum hervorzurufen?	Ja	Das Problem soll verhindert werden. Man könnte zum Beispiel Bedienelemente mit Gültigkeitskontrolle einführen, wenn sie ohne Überprüfung fehleranfällig sind.
Kann das Problem automatisch beseitigt werden?	Ja	Man sollte den Fehler behandeln und dem Nutzer die Fehlermeldung nicht anzeigen.
Ist das Problem während der aktiven Benutzung anderer Anwendungen auch relevant?	Ja	Man sollte die Meldung als Ankündigung darstellen.
Bezieht sich das Problem auf den Status einer Hintergrundaufgabe innerhalb eines primären Fensters?	Ja	Das Problem sollte in der Statusleiste angezeigt werden.

Tabelle 3.2.: Einige Windows-Vorschläge zur Vermeidung der unnötigen Fehlermeldungen, Quelle: [Corb]

Laut den Richtlinien [Core] sollte ein Dialogfeld-Steuerelement für wichtige Informationen, die der Benutzer vor dem Fortsetzen lesen und bestätigen muss, benutzt werden. Ein Fehlerdialogfeld eignet sich besser als ein Inlinefehler, wenn der Fehler für den allgemeinen App-Kontext gilt, beispielsweise im Falle eines Verbindungsfehlers. "Verwenden Sie ein Fragedialogfeld, wenn dem Benutzer eine blockierende Frage gestellt werden muss, z. B. wenn die App nicht im Auftrag des Benutzers eine Auswahl treffen kann," werden weitere Hinweise gegeben, "Eine blockierende Frage kann nicht ignoriert oder verschoben werden und sollte dem Benutzer klar definierte Auswahlmöglichkeiten bieten." [Core]

Aufbau einer Meldung

Ein Windows Dialogfeld ist eine modale Benutzeroberflächenüberlagerung [Corc], die den oberen Teil des Bildschirms überdeckt. Für das Fenster sind minimale sowie maximale Höhen vorgegeben, und es ist mithilfe des "Zurück"-Buttons explizit aufhebbar. Ein Dialogfeld gleicht im Aufbau Android und iOS Meldungen (siehe Seiten 8 bis 9 und 10 bis 12): Es gibt eine Titelbarüberschrift, einen Inhaltstext und die Buttons im Aktionsbereich.

Die Abb. 3.4 zeigt einige Microsoft Design Template Dialogfelder:

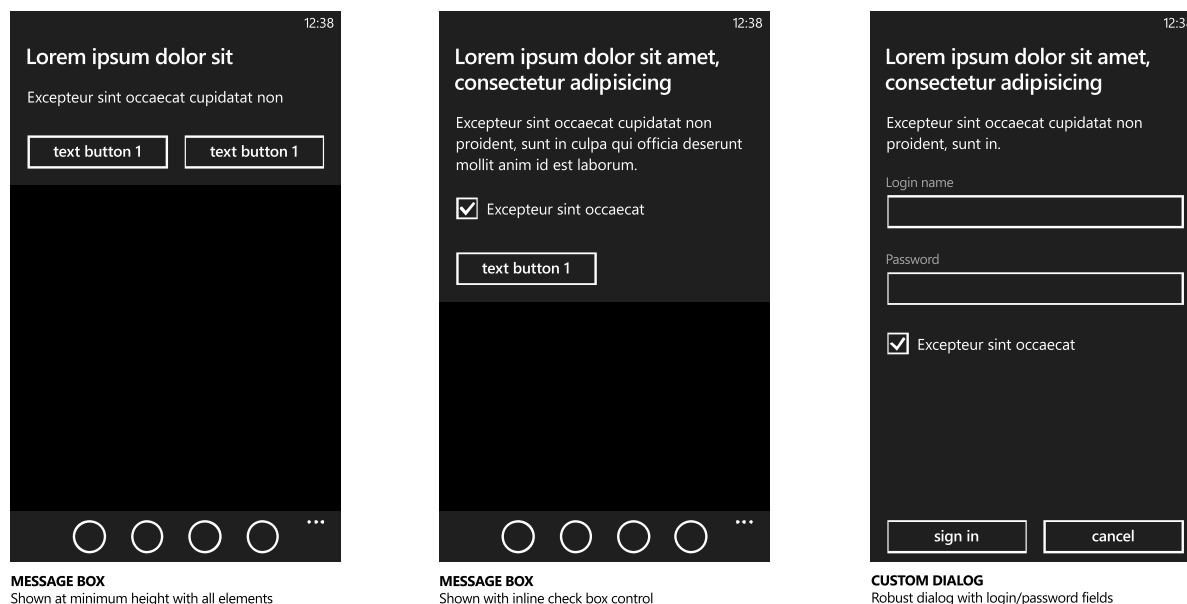


Abbildung 3.4.: Windows Phone 8 Dialogfelder, Quelle: [Cora]

Die Windows 8 Design and coding guidelines fassen die allgemeinen Regeln für die Gestaltung der Meldungen zusammen. Die PDF mit diesen Richtlinien steht auf [Corf] zum Herunterladen bereit.

Als Erstes wird empfohlen: "All message dialogs should clearly identify the user's objective in the first line of the dialog's text." [Cord] Die Hauptanweisung könnte zum Beispiel im optionalen Dialogfeldtitel dargestellt werden. Windows Phone schneidet lange Überschriften ab, daher ist es sinnvoll, diese kurz zu halten. Der Titel soll sich direkt auf die Auswahlmöglichkeiten der Schaltflächen beziehen.

Ein Inhaltsbereich ist im Gegensatz zu dem Titelbereich im Dialogfeld erforderlich. Der Nachrichtentext stellt Details dar. Dabei darf der Titel nicht mit anderen Worten wiederholt werden. Beim Erfassen des Titels und Textes verwendet man die Großschreibung der Satzanfänge und Nomen.

In der Abb. 3.4 ist teilweise zu sehen, dass die Platzierung der Checkboxes sowie Radio-, Toggle- und Druckbuttons im Inhaltsbereich möglich ist. Im Aktionenbereich der Meldung wird mindestens eine Schaltfläche erwartet. Allein stehende Schaltflächen nehmen den linken Platz in der Meldung ein, die Mehrfachbuttons richtet man mittig aus. Die Richtlinien schreiben vor, die zwei nebeneinander stehenden Buttons folgendermaßen zu positionieren:

Specify the default button, which should be the action you most want the user to take.[...]

- If not specified, the default is the leftmost button.
- Put the safest, most conservative choice on the rightmost position.
[Corg]

Die Namen für Schaltflächen werden typischerweise kleingeschrieben. Alle Bezeichnungen sollten für User leicht interpretierbar sein, um ihm bei einer schnellen Entscheidungsfindung zu helfen.

Barrierefreiheit

Um die größtmögliche Zielgruppe zu erreichen, empfehlen die Microsoft Richtlinien [Corg], zahlreiche Fähigkeiten, Einschränkungen und Vorlieben der Anwender zu berücksichtigen. "Folgen Sie von Anfang an den Gestaltungsprinzipien für Barrierefreiheit," [Corg] raten die Herausgeber und listen eine Reihe von Szenarien auf, bei denen eine barrierefreie Applikation funktionieren muss. Diese beziehen sich auf die Sprachausgabe, Tastaturnavigation und visuelle Erfahrung der Nutzer.

Für eine barrierefreie Meldungen sind hierbei folgende Maßnahmen relevant:

- Die Bildschirmsprachausgabe soll Auskunft über die UI-Elementen der Meldung geben. Die Informationen sollen Name, Rolle, Beschreibung, Zustand und Wert beinhalten.
- Die Navigation per Tabulator- und Pfeiltasten soll ermöglicht werden.
- Aktivieren von UI-Elementen mit der Leertaste und der Eingabetaste soll möglich sein.
- Design mit hohem Farbkontrast ist zu bevorzugen.
- Es soll vermieden werden, Informationen nur durch den Einsatz von Farben zu transportieren. Denn die Benutzer, die farbenblind sind, können diese nicht wahrnehmen.

Die Richtlinien weisen außerdem speziell darauf hin, die Erstellung benutzerdefinierter UI-Elemente zu vermeiden, wenn es möglich ist, die Standardsteuerelemente zu verwenden. Denn die "Windows-Runtime-Standardsteuerelemente sind standardmäßig barrierefrei." [Corg]

4

Entwurf der Fehlermeldungen

In diesem Kapitel werden zunächst die ersten Handskizzen (siehe Anhang A.1) und digitalen Mockups zu der "Low battery"-Fehlermeldung auf dem Android-Betriebssystem vorgestellt. Die Dialogstrukturdiagramme unterstützen zum Schluss die Visualisierung der Fehlermeldungen.

4.1. Paper Prototyping

Nach Vorgabe [PMR14] verfügt die imaginäre Geschäftsprozess-Software über zwei Sichten: Worklist-View und Execution-View. In der Arbeitslistenansicht haben die Nutzer die Möglichkeit, die im System vorhandenen Daten anzuschauen, auszuwählen, nicht aber zu manipulieren. Die unmittelbare Datenbearbeitung geschieht in der Ausführungsansicht. Dabei wird dem User ein Formular zur Verfügung gestellt, in das die zu speichernden Informationen einzutragen sind. Die Formulardaten werden nach jeder Eingabe auf dem Server gespeichert. In beiden Views können die gleichen Probleme auftreten, die Benachrichtigungen darüber sollten dennoch individuell angepasst werden.

Folgende Fehlerfälle kamen in Betracht (siehe Anhang A.1):

- niedriger Akkuladestand,
- nicht verfügbare Netzwerkverbindungen,
- undefinierter Benutzerstandort,
- sofortiges Ausschalten des Gerätes,
- Zeitbegrenzung.

Um Ideen und Abläufe möglichst früh prüfen zu können, wurde in der Anfangsphase der Konzeption eine Methode der nutzerorientierten Gestaltung namens Paper Prototyping eingesetzt. Bei dieser transparenten Darstellungsweise sind die Änderungen am schnellsten vornehmbar. Nach und nach entstanden auf den gedruckten Android-Bildschirmen von Hand skizzierte GUI-Komponenten (siehe Abb. 4.1 und Abb. 4.2). Aus Gründen der Weiterverwendung wurden alle Meldungen der Prozess-App in Englisch verfasst.

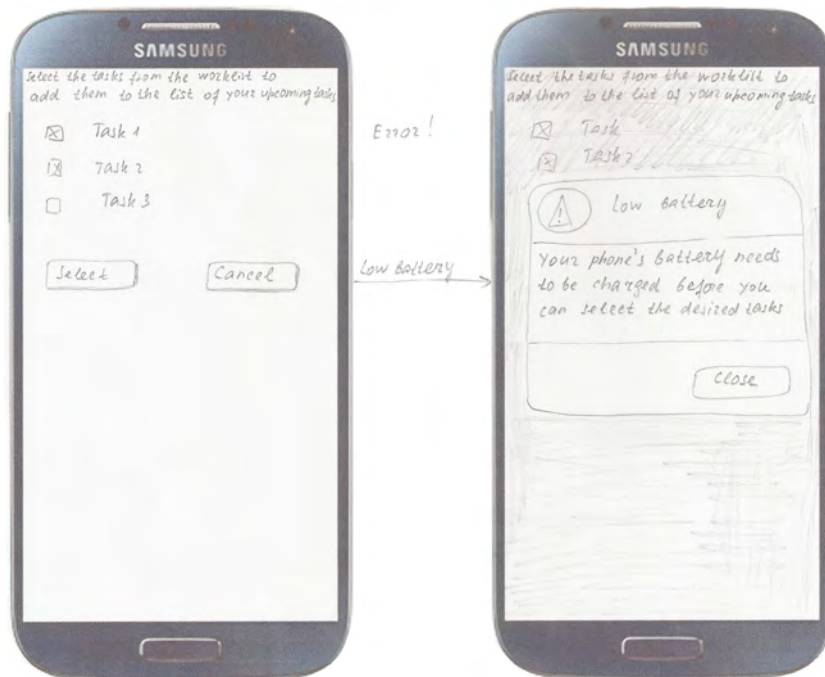


Abbildung 4.1.: Skizze des "Task Selection"-Dialogs und der "Low battery"-Fehlermeldung in der Worklist-View

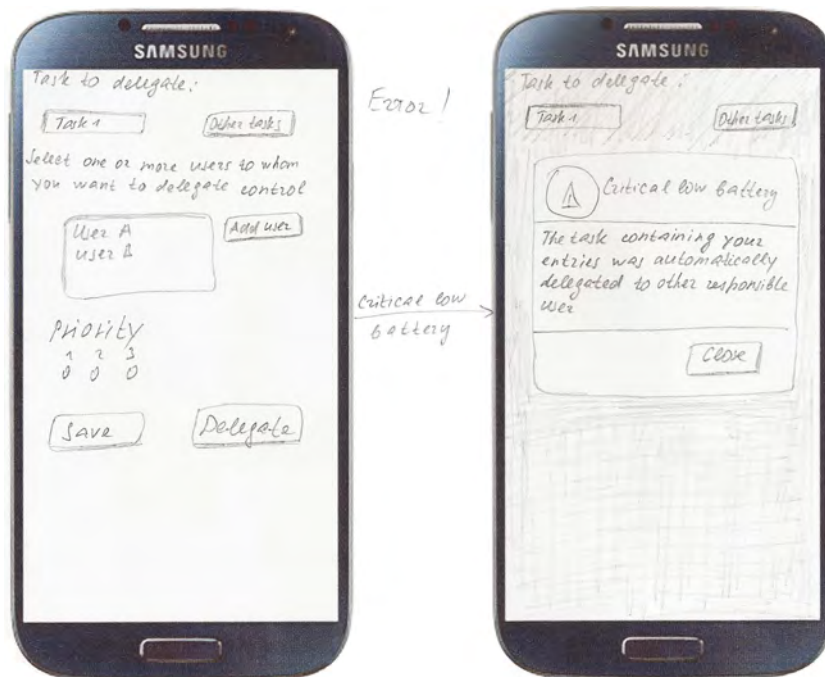


Abbildung 4.2.: Skizze des "Task Delegation"-Dialogs und der "Critical low battery"-Fehlermeldung in der Execution-View

Bei weniger dringlichen Aufgaben gibt das System im Fehlerfall eine einfache Informationsmeldung aus. Die Prozesse, die jedoch sofort auszuführen sind, werden delegiert. Hierbei entstehen komplizierte Fehlermeldungen.

4.2. Dialogbeschreibung

Basierend auf den Handskizzen begann im nächsten Schritt die Modellierung digitaler Mockups. Alle elektronischen Entwürfe wurden mit Adobe Photoshop CS6 [Ltd] erstellt. Hierfür waren die vorgefertigten Vorlagen von Android [LLC] [Inc] und Windows Phone [Cora] hilfreich.

Die Unterkapitel 4.2.1 und 4.2.2 stellen die Ausgangsdialoge sowie die Benachrichtigungen, die im Fall des niedrigen Akkustandes in Erscheinung treten, vor. Die Mockups zu weiteren Fehlerfällen befinden sich im Anhang A.2.

4.2.1. Worklist-View

Genauere Betrachtung verdienen hier die Meldungen, die während der Auswahl, Ausführung und Delegation der Aufgaben in der Arbeitsliste angezeigt werden.

4.2.1.1. Task Selection

Dialog "Task Selection"

Der Dialog "Task Selection" (siehe Abb. 4.3) wird angezeigt, wenn der Benutzer eine bzw. mehrere der verfügbaren Aufgaben zu seiner Aufgabenliste hinzufügen will.

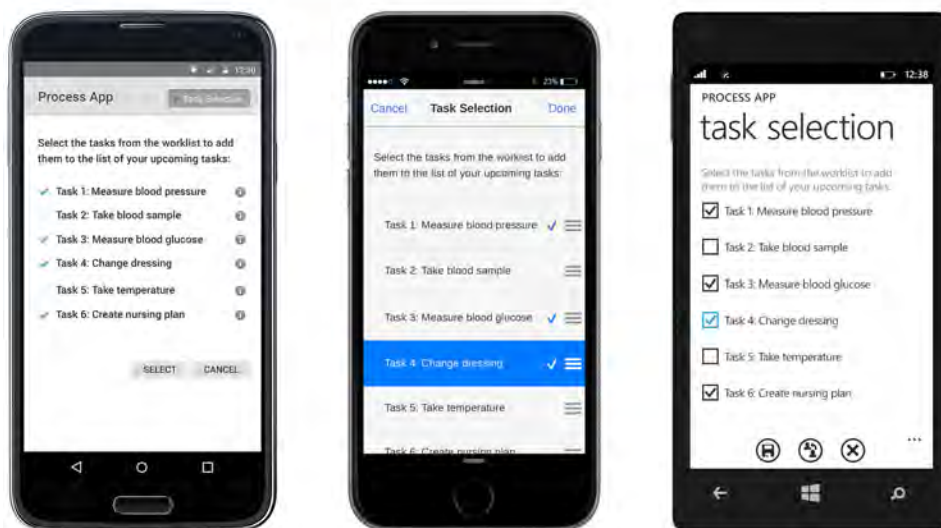


Abbildung 4.3.: Dialog "Task Selection" - Android, iOS, Windows Phone

Dialog "Low battery" während der Aufgabenauswahl

Das "Low battery"-Mitteilungsfenster erscheint während der Aufgabenauswahl, falls der Akku des Smartphones fast leer ist. Sie warnt den Benutzer: "Your phone's battery needs to be charged before you can select the desired tasks." Möchte der User die Fehlermeldung schließen, kann er dies mit dem "Close"-Button tun. Nach dieser Interaktion wird der vorherige Dialog "Task Selection" geöffnet (siehe Abb. 4.4 und Abb. 4.3).

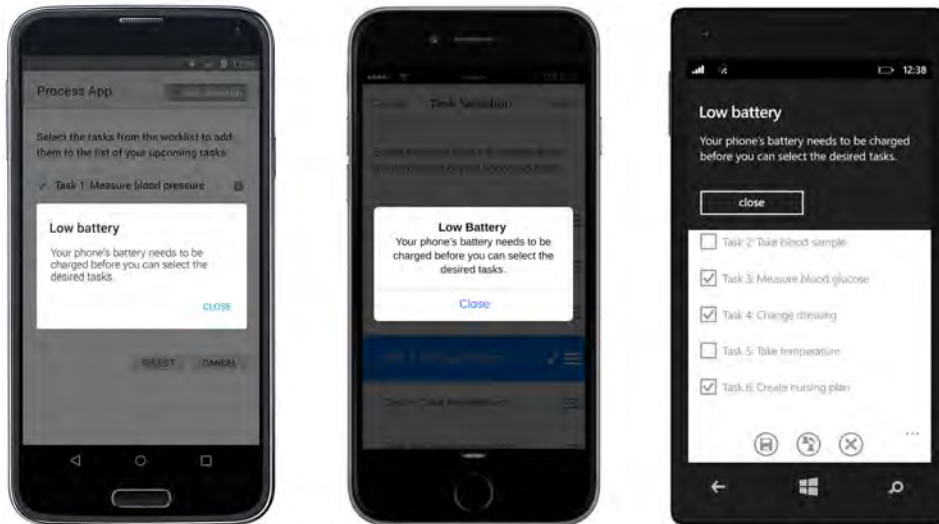


Abbildung 4.4.: Dialog "Low battery" während der Aufgabenauswahl - Android, iOS, Windows Phone

4.2.1.2. Task Execution

Dialog "Task Execution"

Will der Benutzer eine Aufgabe aus seiner Aufgabenliste ausführen, kann er das im Dialog "Task Execution" (siehe Abb. 4.5) erledigen.

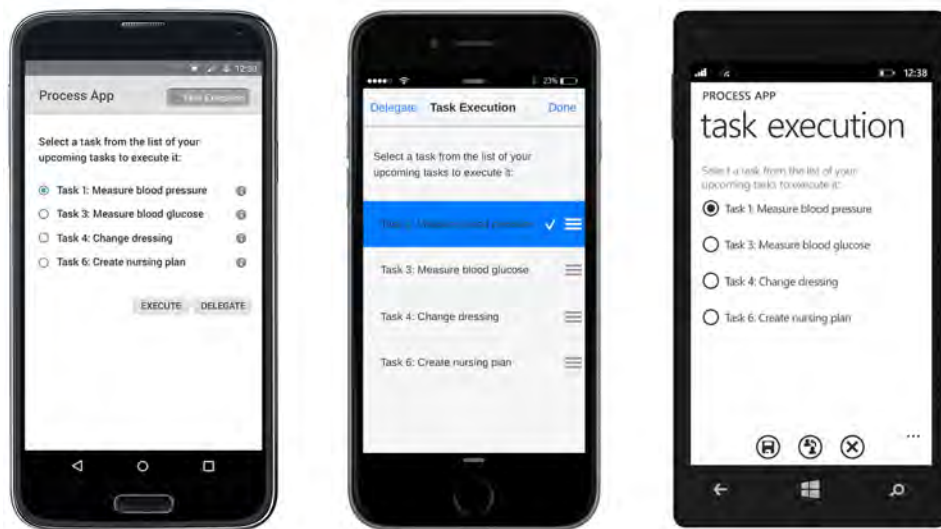


Abbildung 4.5.: Dialog "Task Execution" - Android, iOS, Windows Phone

Dialog "Low battery" während der Aufgabenausführung

"Your phone's battery needs to be charged before task execution." - Diese "Low battery"-Fehlermeldung wird dem Benutzer während der Aufgabenausführung gezeigt. Er kann die Fehlermeldung mit der Schaltfläche "Close" schließen, um in den vorherigen Dialog für die Aufgabenausführung zu gelangen (siehe Abb. 4.6 und Abb. 4.5).

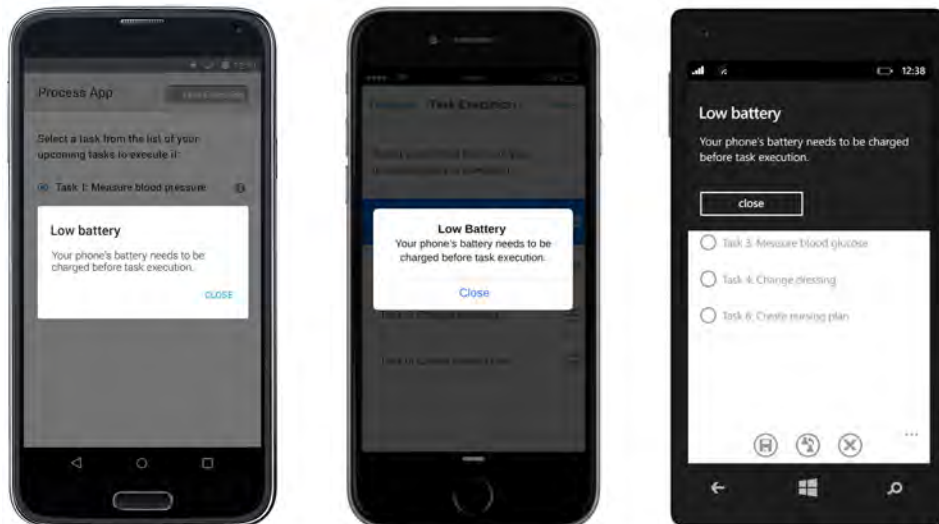


Abbildung 4.6.: Dialog "Low battery" während der Aufgabenausführung - Android, iOS, Windows Phone

4.2.1.3. Task Delegation

Dialog "Task Delegation"

Die Delegation einer bzw. mehreren Aufgaben an die anderen Benutzer wird im Dialog "Task Delegation" (siehe Abb. 4.7) durchgeführt.

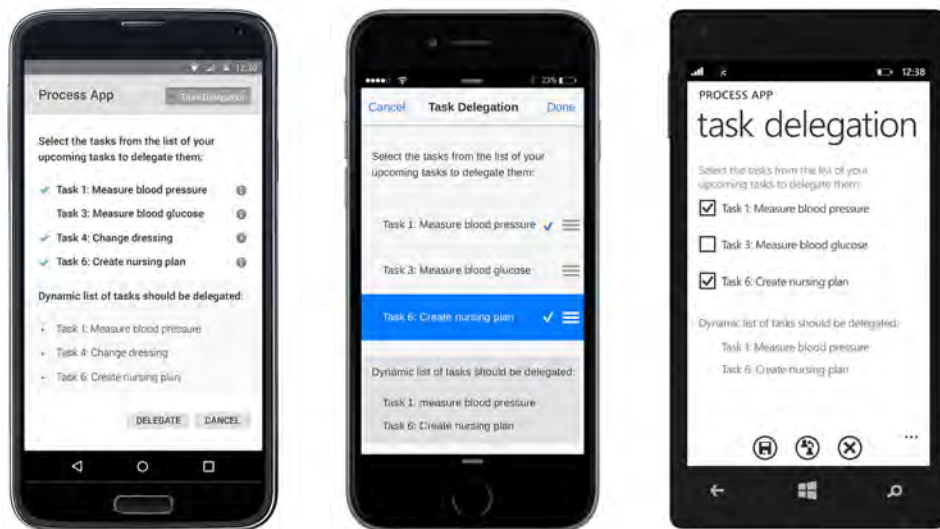


Abbildung 4.7.: Dialog "Task Delegation" - Android, iOS, Windows Phone

Dialog "Low battery" während der Aufgabendelegation

Falls der Akku des Smartphones während der Aufgabendelegation beinahe erschöpft ist, wird der Benutzer zunächst nach dem Speichern der Delegationsliste gefragt: "Save the list of tasks should be delegated?" Der User kann die für Delegation vorgesehenen Aufgaben mit dem "Save"-Button speichern (siehe Abb. 4.8). Ein neuer Dialog bestätigt die Speicherung (siehe Abb. 4.9). Alternativ kann die Fehlermeldung mit dem "Close"-Button geschlossen werden. In diesem Fall landet man im Dialog "Task Delegation" (siehe Abb. 4.7). Die aktuelle Delegationsliste geht verloren.

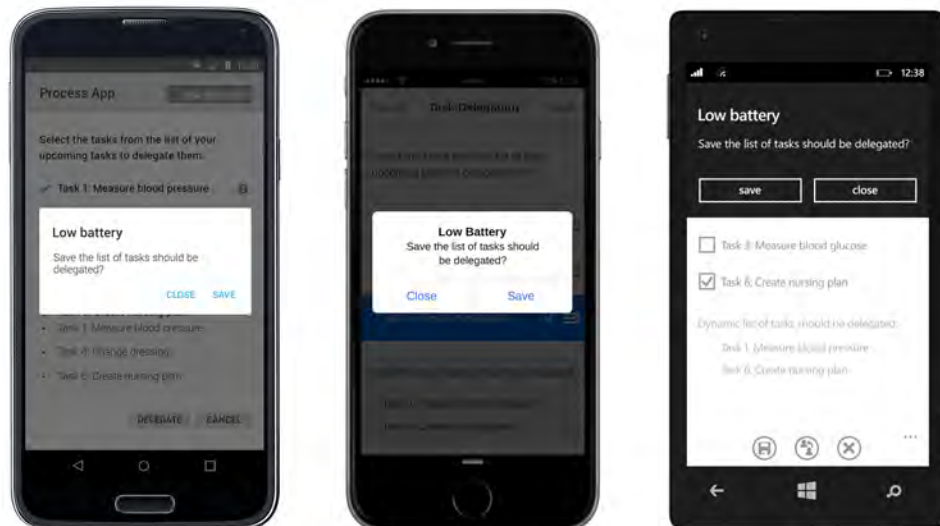


Abbildung 4.8.: Dialog "Low battery" während der Aufgabendelegation - Android, iOS, Windows Phone

Bestätigungsdialog "Low battery" während der Aufgabendelegation

Nach dem Anklicken des Buttons "Save" im vorherigen Dialog (siehe Abb. 4.8) wird das Speichern bestätigt (siehe Abb. 4.9). Anschließend folgt die Erinnerung, dass der Akku des Geräts geladen werden muss: "The list of tasks should be delegated was saved. Your phone's battery needs to be charged before task delegation." Zum Schließen des Dialogs wird ein "Close"-Button angeboten, der zum Dialog "Task Delegation" zurückführt (siehe Abb. 4.7).

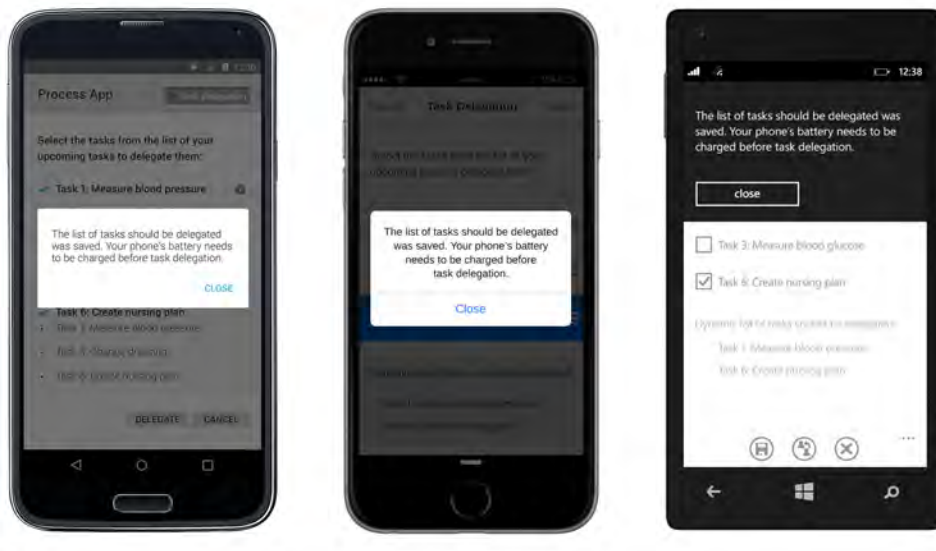


Abbildung 4.9.: Bestätigungsdialog "Low battery" während der Aufgabendelegation - Android, iOS, Windows Phone

4.2.2. Execution-View

Die in diesem Unterkapitel beschriebenen Meldungen erscheinen während der Auswahl, Ausführung und Delegation der Aufgaben in der Ausführungsansicht.

4.2.2.1. Task Execution

Dialog "Task Execution"

In der Ausführungssicht ist der Dialog "Task Execution" ein Formular, in dem die für die Aufgabenausführung notwendigen Daten eingegeben werden (siehe Abb. 4.10).

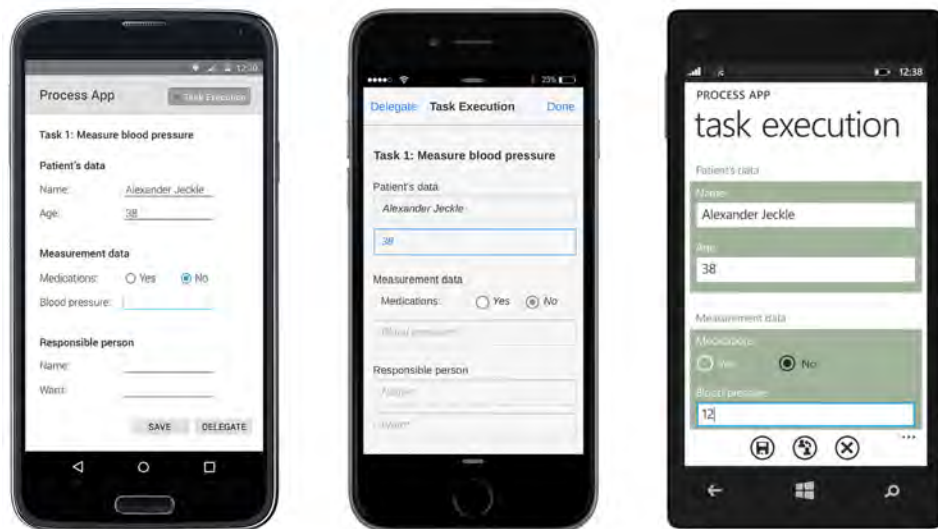


Abbildung 4.10.: Dialog "Task Execution" - Android, iOS, Windows Phone

Dialog "Low battery" während der Aufgabenausführung (Stufe 1)

Wie am Anfang dieses Kapitels bereits erwähnt werden die Formulardaten nach jeder Eingabe gespeichert. Das System teilt es dem Benutzer mit, wenn der Akku des Geräts während des Ausfüllens schwach wird (siehe Abb. 4.11). Des Weiteren bekommt der User einen Hinweis zur Akku-Ladung: "The already entered values were saved. Your phone's battery needs to be charged". Ein Klick auf die Schaltfläche "Close" lässt die Fehlermeldung vom Bildschirm verschwinden. Danach erscheint erneut das Formular der Aufgabenausführung (siehe Abb. 4.10).

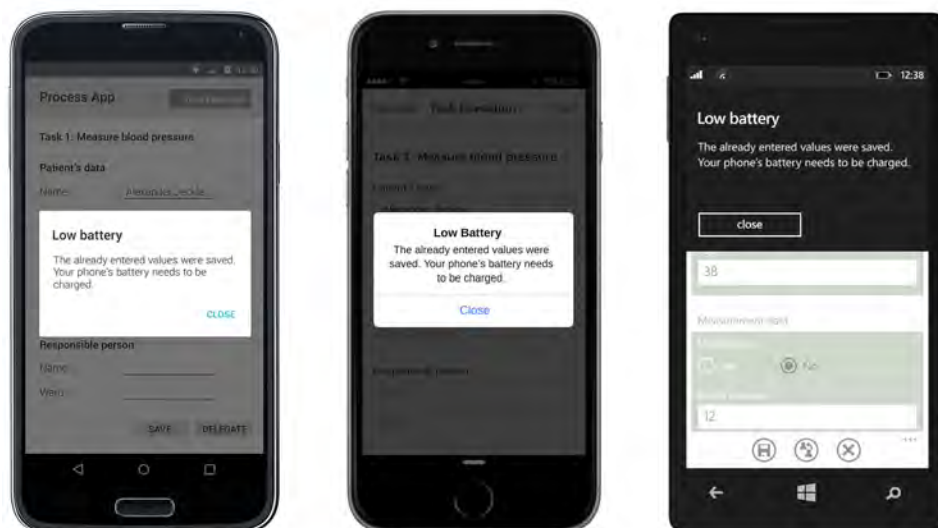


Abbildung 4.11.: Dialog "Low battery" während der Aufgabenausführung - Android, iOS, Windows Phone

Dialog "Low battery" während der Aufgabenausführung (Stufe 2)

Wurde das Smartphone noch nicht an ein Netzteil angeschlossen, bekommt der Benutzer eine erneute Erinnerung den Akku zu laden (siehe Abb. 4.12). Da das Gerät ohne Energieversorgung bald ausgeschaltet wird, fragt das System, ob die noch nicht erledigte Aufgabe an einen anderen Verantwortlichen weitergeleitet werden soll: "Your phone's battery needs to be charged immediately. Would you like to delegate the task containing your entries to a certain responsible user?". Für diese Aktion steht der Button "Delegate" bereit. Ein Klick darauf führt zum Formular der Aufgabendelegation (siehe Abb. 4.14). Möchte der Benutzer die Aufgabe nicht delegieren, kann er die Fehlermeldung mit dem "Close"-Button schießen. Es wird das Formular der Aufgabenausführung angezeigt (siehe Abb. 4.10).

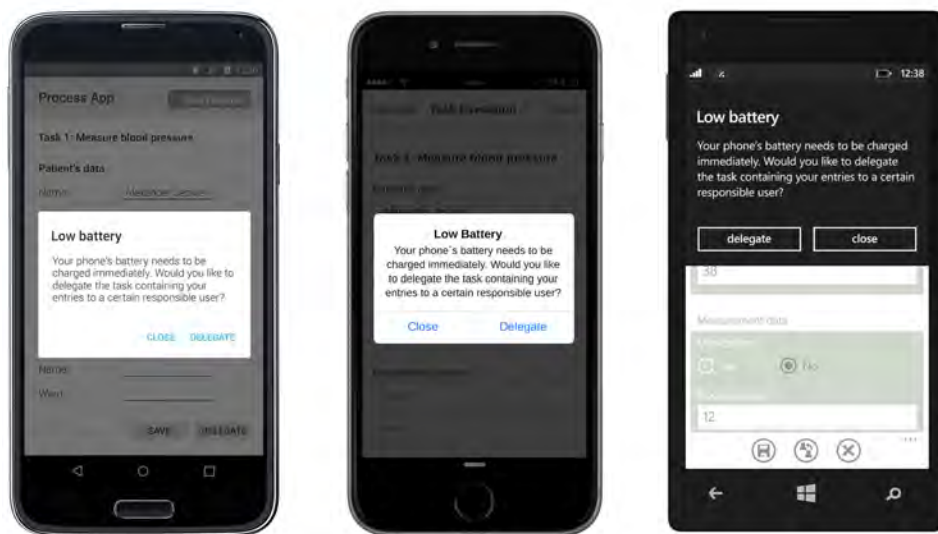


Abbildung 4.12.: Dialog "Low battery" während der Aufgabenausführung - Android, iOS, Windows Phone

Dialog "Critical low battery" während der Aufgabenausführung (Stufe 3)

Wenn die Akkukapazität das kritische Niveau erreicht, öffnet sich der Dialog "Critical low battery" (siehe Abb. 4.13). Diesmal wird dem Benutzer mitgeteilt, dass das noch nicht zu Ende ausgefüllte Formular mit allen gespeicherten Eingaben an einen anderen Verantwortlichen automatisch delegiert wurde: "The task was saved and automatically delegated to other responsible user. You can check if it is available in the list of your upcoming tasks after recharging the phone's battery." Falls die Aufgabe nach dem Laden des Akkus von keinem anderen übernommen wurde, steht sie dem Benutzer in der Liste seiner bevorstehenden Aufgaben (siehe Abb. 4.3) wieder zur Verfügung. Ein "Close"-Button schießt die Fehlermeldung. Das Formular der Aufgabenausführung tritt in Erscheinung (siehe Abb. 4.10).

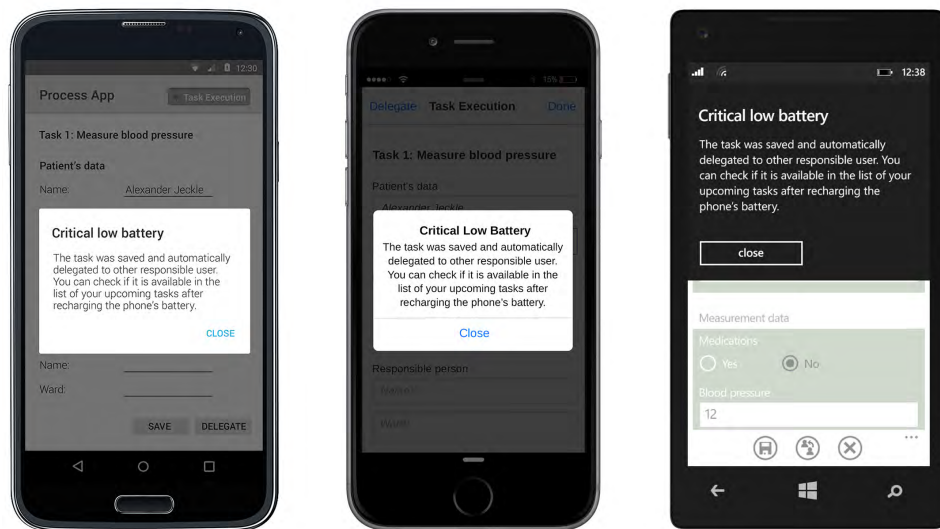


Abbildung 4.13.: Dialog "Critical low battery" während der Aufgabenausführung - Android, iOS, Windows Phone

4.2.2.2. Task Delegation

Dialog "Task Delegation"

Um eine bestimmte Aufgabe an einen bzw. mehrere Verantwortliche(n) weiterzuleiten, soll der Benutzer ein Formular für die Aufgabendelegation ausfüllen (siehe Abb. 4.14).

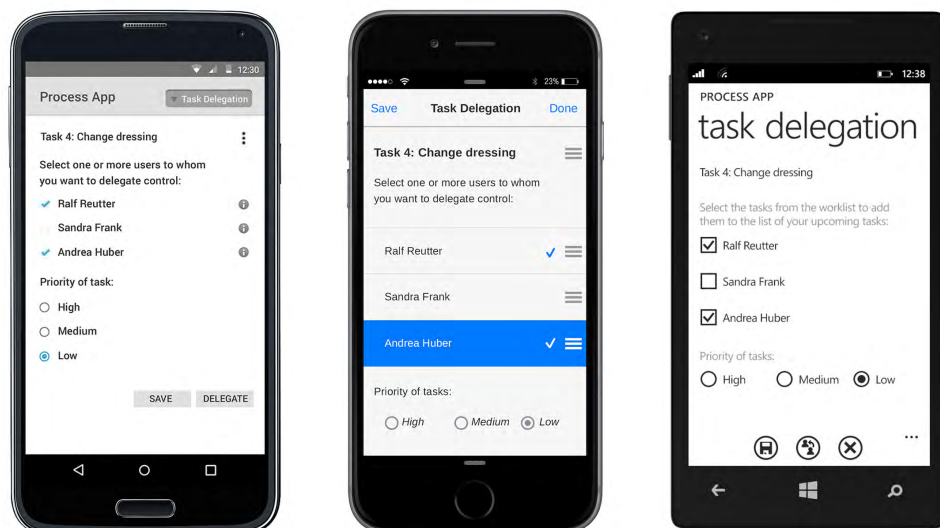


Abbildung 4.14.: Dialog "Task Delegation" - Android, iOS, Windows Phone

Dialog "Low battery" während der Aufgabendelegation (1. Stufe)

Im Dialog "Task Delegation" geschieht das Speichern der Formulardaten ebenso feldweise. Bei einem niedrigen Akkustand informiert das Gerät den User, dass die erfassten Inhalte gespeichert wurden, und weist auf die Notwendigkeit hin, den Akku zu laden: "The already entered values were saved. Your phone's battery needs to be charged". Die Fehlermeldung lässt sich mit der Schaltfläche "Close" schließen (siehe Abb. 4.15). Der Benutzer sieht wieder das Formular für die Aufgabendelegation (siehe Abb. 4.14).

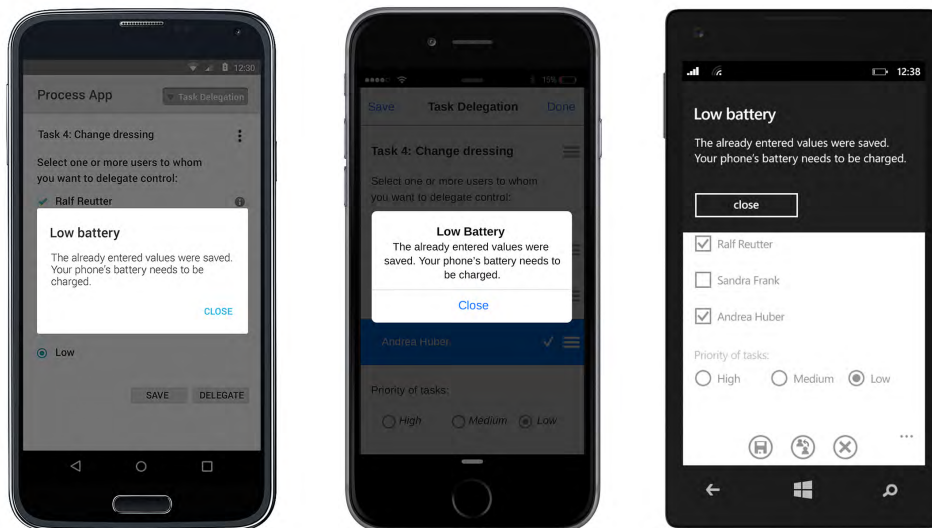


Abbildung 4.15.: Dialog "Low battery" während der Aufgabendelegation - Android, iOS, Windows Phone

Dialog "Critical low battery" während der Aufgabendelegation (2. Stufe)

Der Benutzer erhält diese Fehlermeldung (siehe Abb. 4.16) während der Aufgabendelegation, wenn die Akkuladung für die Funktionstüchtigkeit des Geräts zu niedrig ist. Das System führt eine automatische Delegation der zu delegierenden Aufgabe durch und meldet sich: "The task containing your entries was automatically delegated to other responsible user." Das Schließen des Dialogs ist durch ein Klick auf den "Close"-Button möglich. Es öffnet sich der vorherige Dialog "Task Delegation" (siehe Abb. 4.14).

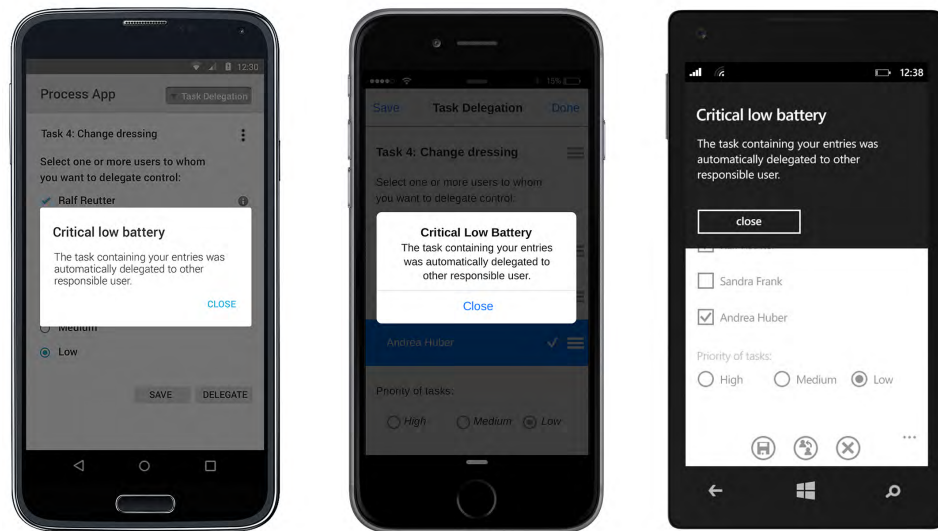


Abbildung 4.16.: Dialog "Critical low battery" während der Aufgabendelegation - Android, iOS, Windows Phone

4.3. Dialogstrukturdiagramme

Zum Zwecke der besseren Strukturierung ist die Abfolge der oben beschriebenen Dialoge in Abhängigkeit der betätigten Buttons und der auftretenden Ereignisse graphisch aufbereitet (siehe Abb. 4.17 und Abb. 4.18). In beiden Sichten (Worklist- und Execution-View) sind Fehlermeldungen als modale Dialogfenster dargestellt. "Das bedeutet, dass so lange kein anderes Fenster der Anwendung aktiviert werden kann, bis das modale Fenster geschlossen wird". [Kue] Der Benutzer kann erst fortfahren, wenn er auf die Mitteilung reagiert hat.

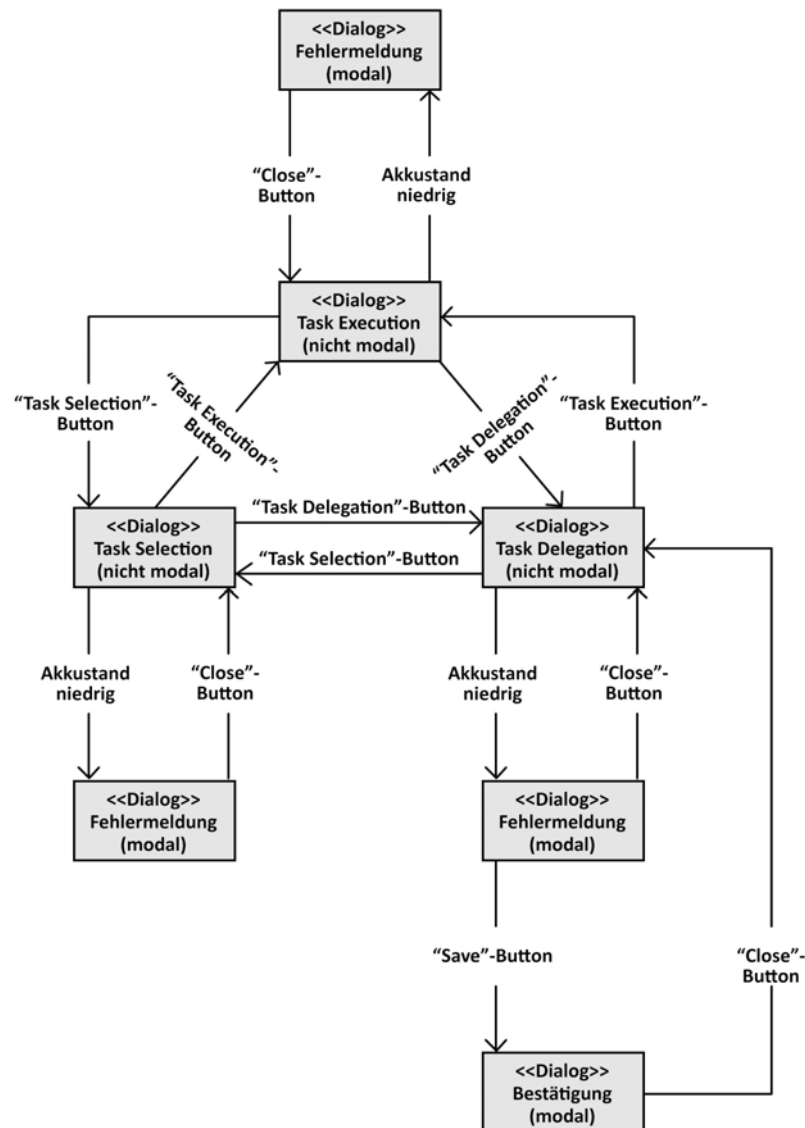


Abbildung 4.17.: Dialogstrukturdiagramm "Low battery" - Worklist-View

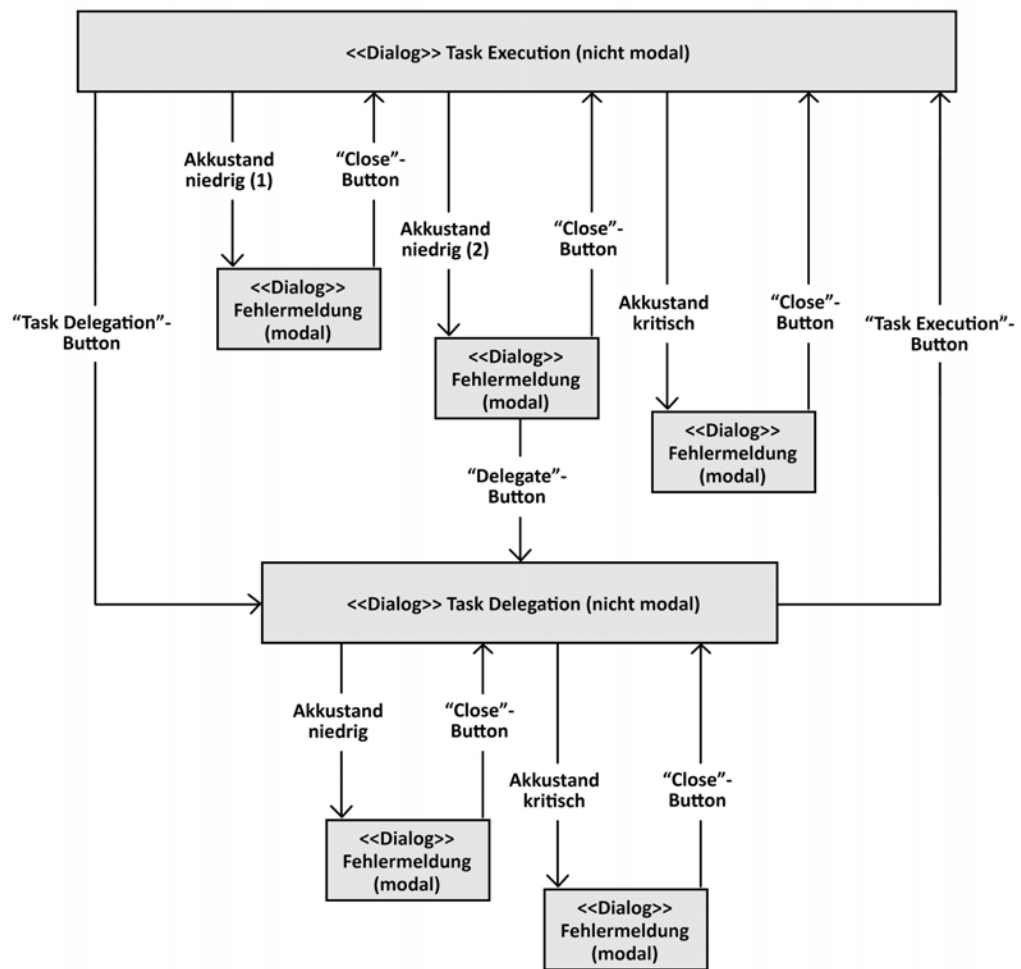


Abbildung 4.18.: Dialogstrukturdiagramm "Low battery" - Execution-View

5

Umfrage

Im Folgenden wird die mit Hilfe des Internetanbieters "Umfrage Online" [Gmb] durchgeführte Umfrage ausdiskutiert. Nach der Schilderung der Ziele soll ein Blick auf den Aufbau der Fragebögen geworfen werden. Des Weiteren kommt die Auswertung der gewonnenen Ergebnisse. Die von den Teilnehmern vorgeschlagenen Verbesserungsmöglichkeiten schließen das Kapitel ab.

5.1. Ziel der Umfrage

Bei der Ausführung von Prozess-Aktivitäten auf Smart-Mobilgeräten können die unterschiedlichsten Fehler auftreten, mit denen umgegangen werden muss. Im Rahmen dieser Arbeit wurden Fehlermeldungen für einige Anwendungsfälle ausgearbeitet. Jede Meldung sollte dem Nutzer eine möglichst klare Aussage über das aktuelle Problem und seine Lösung vermitteln. Außerdem stehen optimales Layout, passende Textlänge, eindeutige Benennung der Buttons und verständlicher Dialogablauf im Blickfeld. Ziel dieser Online-Umfrage war herauszufinden, ob das entwickelte Konzept die Zufriedenheit der Teilnehmer in den oben genannten Punkten gewährleistet, und mögliches Verbesserungspotenzial zu identifizieren.

5.2. Aufbau der Umfrage

Nach einigen allgemeinen Fragen zur Person wurden die Aufgabenauswahl-, Aufgabenausführungs- und Aufgabendelegations-Mockups einer beispielhaften Android-Anwendung eingeführt. Jedem Dialog folgte eine Meldung des "Low battery"-Fehlerfalls (siehe Abb. 5.1 und Abb. 5.2).

Die Befragten sollten die Optik und die Verständlichkeit der Fehlermeldungen mittels einer Likert-Skala bewerten. Dabei handelte es sich um bestimmte Aussagen und die bereits im Voraus bestehenden Kategorisierung der Antworten. Durch die Gestaltung der einheitlichen Auswahlmöglichkeiten erzielt diese Befragungsmethode eine erhöhte Vergleichbarkeit in der Auswertung. Zum Schluss konnten die Teilnehmer angeben, wie hilfreich die jeweilige Mitteilung beim Lösen des Problems war, und die eigene Verbesserungsvorschläge als Freitext einbringen. Abbildung 5.3 zeigt einen Fragebogenauszug. Die komplette Umfrage mit Auswertung befindet sich im Anhang.

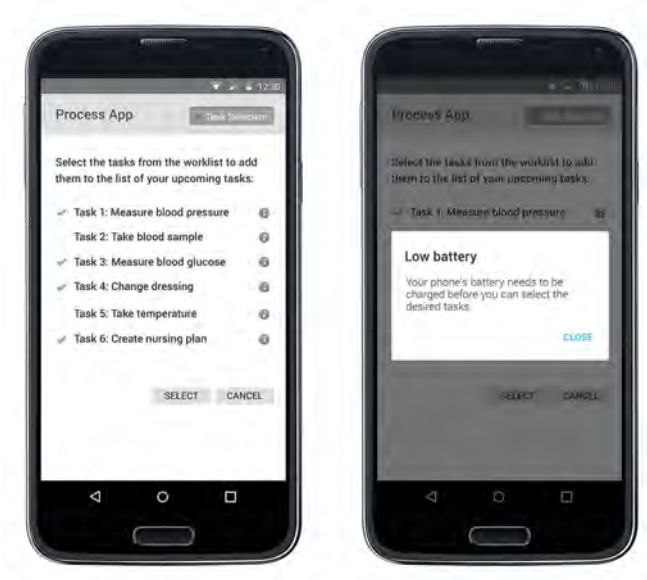


Abbildung 5.1.: Dialog der Aufgabenauswahl und dazugehörige Fehlermeldung "Low battery"

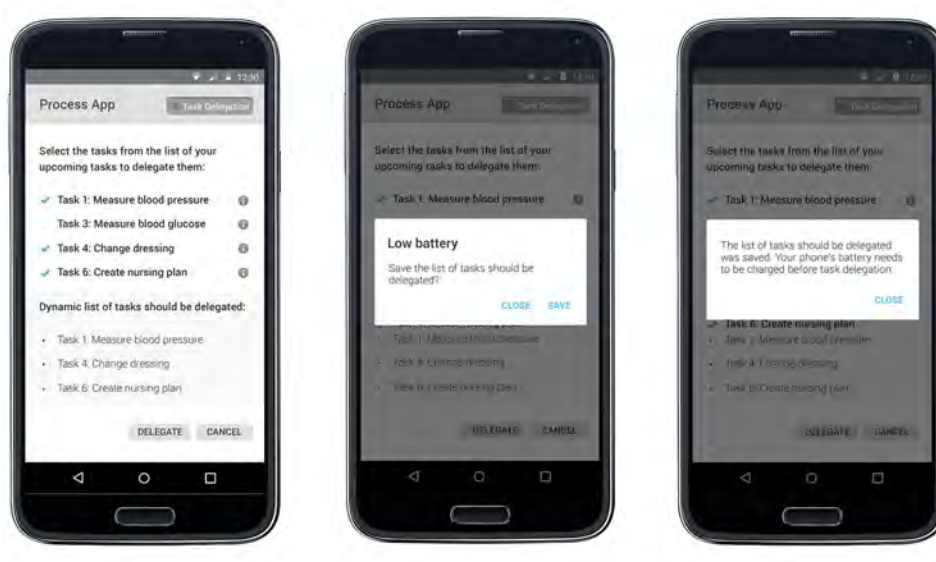


Abbildung 5.2.: Dialog der Aufgabendelegation und dazugehörige Fehlermeldung "Low battery"

Die Formulierung der Fehlermeldung ist gut verständlich.

☐ trifft zu ☐ trifft eher zu ☐ teils-teils ☐ trifft eher nicht zu ☐ trifft nicht zu

Die Fehlermeldung hebt sich gut vom Hintergrund ab.

☐ trifft zu ☐ trifft eher zu ☐ teils-teils ☐ trifft eher nicht zu ☐ trifft nicht zu

Die Benennung des Buttons "Close" ist eindeutig.

☐ trifft zu ☐ trifft eher zu ☐ teils-teils ☐ trifft eher nicht zu ☐ trifft nicht zu

Sie verstehen sehr gut, was passieren würde, wenn Sie den Button "Close" betätigen.

☐ trifft zu ☐ trifft eher zu ☐ teils-teils ☐ trifft eher nicht zu ☐ trifft nicht zu

Abbildung 5.3.: Fragen bezüglich der "Low battery"-Fehlermeldung während der Aufgabenauswahl

5.3. Ergebnisse der Umfrage

5.3.1. Auswertung

Die Fragebögen wurden von "Umfrage Online" [Gmb] automatisch ausgewertet. Insgesamt nahmen 23 Personen im Durchschnittsalter von 25-34 Jahren an der Umfrage teil. Knapp 70 Prozent der Befragten waren männlich. Die Mehrheit studierte Informatik bzw. Medieninformatik an der Universität Ulm und benutzte das Android-Betriebssystem (siehe Abb. 5.4 und Abb. 5.5).

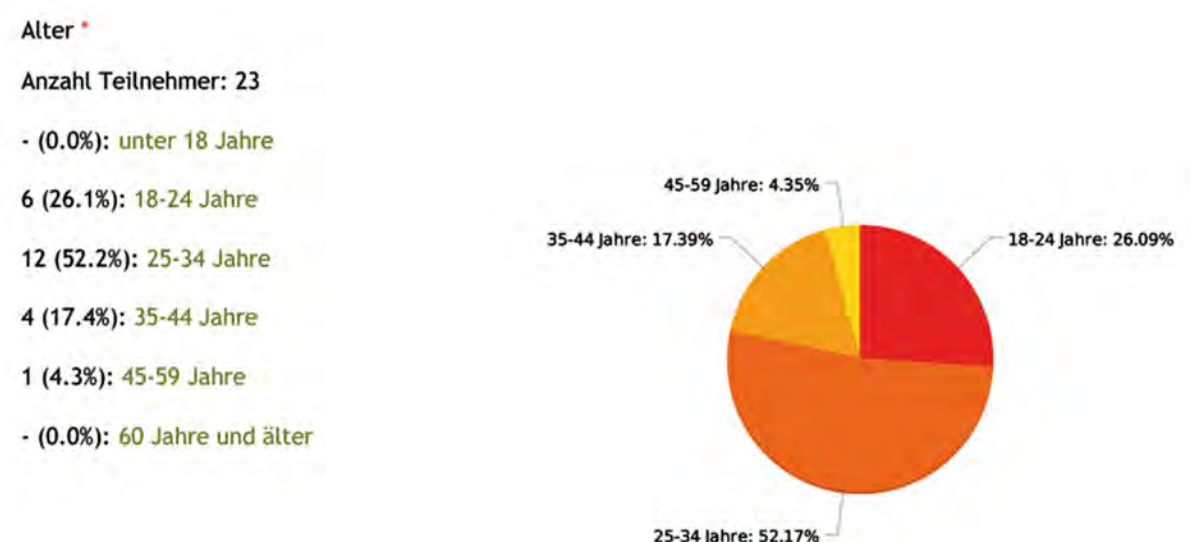


Abbildung 5.4.: Auswertung zur allgemeinen Frage über das Alter der Teilnehmer

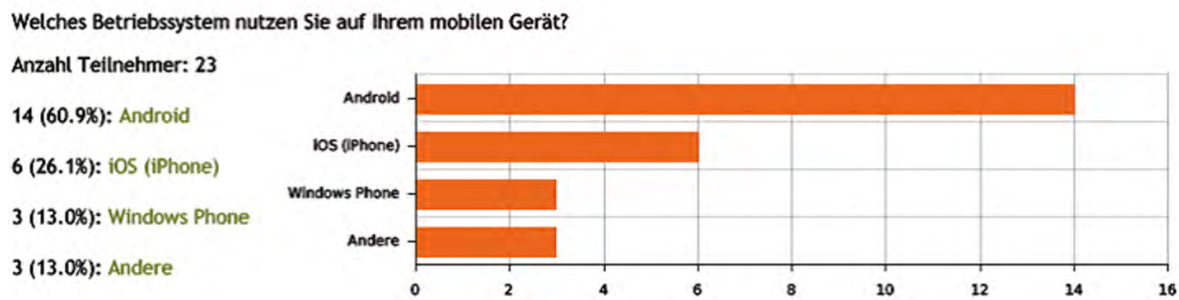


Abbildung 5.5.: Auswertung zur allgemeinen Frage über das meist benutzte Betriebssystem

Rund 4,3 Prozent der Teilnehmer finden die Fehlermeldungen im Allgemeinen eher nicht nützlich. Genau so viele befolgen die Anweisungen in der Regel überhaupt nicht und 8,7 Prozent machen es selten. Nichts desto trotz liest mehr als die Hälfte die Systemmitteilungen, um die Hintergründe der aufgetretenen Fehler zu erfahren und sie zu beheben (siehe Abb. 5.6).

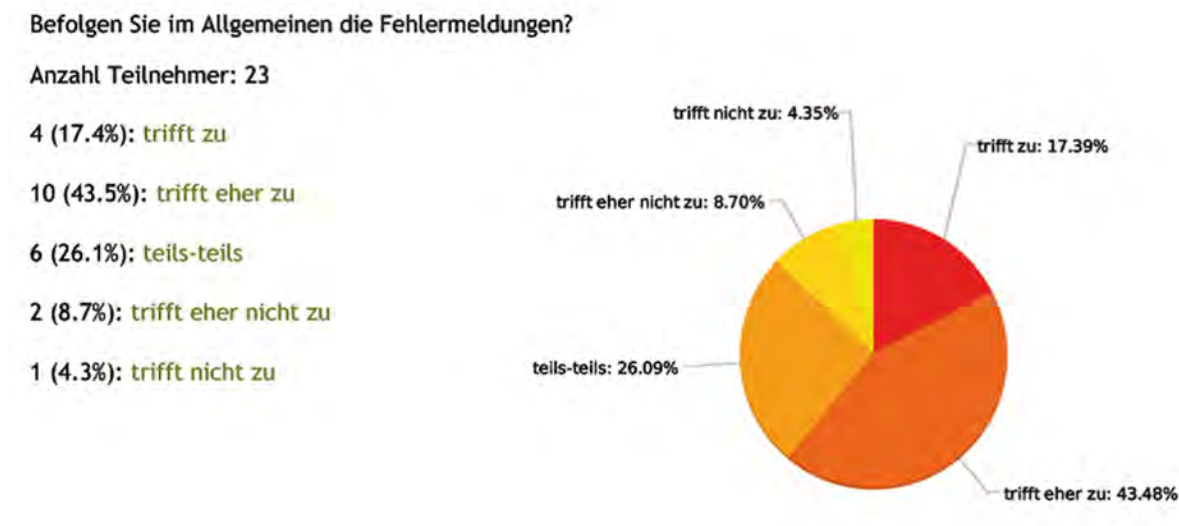


Abbildung 5.6.: Auswertung zur allgemeinen Frage über das Befolgen der Fehlermeldungen

Fast alle Befragten bewerteten das Layout der im Rahmen dieser Arbeit entstandenen Mockups positiv. Die Schriftgröße, der Wort- und Zeilenabstand wurden größtenteils als optimal empfunden. Die Mehrheit bestätigte, dass die Fehlermeldungen sich gut vom Hintergrund abheben. Unstimmigkeiten gab es allerdings bezüglich der Textlänge (siehe Abb. 5.7).

Der Text der Fehlermeldung ist zu lang.

Anzahl Teilnehmer: 23

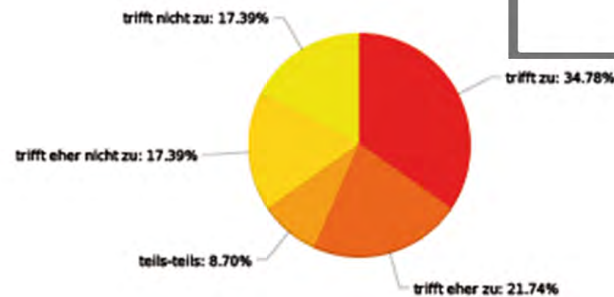
8 (34.8%): trifft zu

5 (21.7%): trifft eher zu

2 (8.7%): teils-teils

4 (17.4%): trifft eher nicht zu

4 (17.4%): trifft nicht zu



Critical low battery

The task was saved and automatically delegated to other responsible user. You can check if it is available in the list of your upcoming tasks after recharging the phone's battery.

CLOSE

Abbildung 5.7.: Auswertung zur Frage über die Textlänge

Die Formulierung der Meldungen, die lediglich die Ursache des Problems vermitteln und mit einem "Close"-Button zu schließen sind, fanden die Nutzer verständlich. Auch das Delegation-Prinzip war für die Meisten gut nachvollziehbar. Unsicherheiten gab es hingegen bei den Fehlermeldungen zur Fortschrittsspeicherung (siehe Abb. 5.8).

Die Formulierung der Fehlermeldung ist gut verständlich.

Anzahl Teilnehmer: 23

7 (30.4%): trifft zu

4 (17.4%): trifft eher zu

4 (17.4%): teils-teils

7 (30.4%): trifft eher nicht zu

1 (4.3%): trifft nicht zu



Low battery

Save the list of tasks should be delegated?

CLOSE SAVE

Abbildung 5.8.: Auswertung zur Frage über die Fortschrittsspeicherung

5.3.2. Kommentare der Befragten

Wie bereits im Kapitel 5.2 erwähnt, hatten die Teilnehmer der Umfrage die Möglichkeit, die Anregungen hinsichtlich der Verbesserung der Fehlermeldungen zu hinterlassen. Da die Meinung der Benutzer eine entscheidende Rolle für ein Usability-Konzept spielt, sollte an dieser Stelle

auf die relevanten Anmerkungen der Befragten eingegangen werden. Alle Kommentare und Verbesserungsvorschläge sind im vollen Umfang im Anhang A.3 aufgeführt.

5.3.2.1. Worklist-View

Fehlermeldung "Low battery" während der Aufgabenauswahl

Je nach Studiengang variierten die Meinungen zu dem Inhalt und der Textlänge erheblich. Während die "Nicht-Informatik"-Studenten die vorgestellte Mitteilung (siehe Kapitel 4.2.1.1, Abb. 4.3 und Abb. 4.4 als klar bezeichneten, hatten die Studierenden der Informatikstudiengänge andere Erwartungen. So benötigte eine Wirtschaftsmathematikerin keine zusätzlichen Informationen über den minimalen bzw. aktuellen Akkuzustand, um eine Aufgabe zu wählen. Ihrer Ansicht nach, wirkt eine vertiefte Beschreibung eher irritierend und zerstört die Einfachheit und äußere Struktur der Fehlermeldung. Dafür interessierten sich zwei Medieninformatiker und ein Software Ingenieur für die weiteren Details zu dem aufgetretenen Fehler. Einer von ihnen schrieb: "Für mich wäre eine Begründung oder eine Möglichkeit, um sich zu diesem Problem mehr Details einblenden zu lassen, ganz gut". Ein anderer wollte wissen, ab welchem Akkustand sein Smartphone wieder betriebsfähig ist, oder ob er nach dem Laden den Service weiter benutzen kann. Interessanterweise würde ein Medieninformatiker die Beschreibung kürzer halten.

Die Bedeutung des Buttons "Close" verwirrte manche Umfrageteilnehmer: "Schließt close die Fehlermeldung oder die App?" Nach dem Lesen der Überschrift "Low Battery" wanderte der Blick eines Wirtschaftsphysikers sofort auf den "Close"-Button. Als mögliche Verbesserung schlug er ein "X"-Button an der oberen rechten Ecke anstatt dem "Close"-Button vor: "... oder sogar gar keinen Button zum Schließen verwenden". Im letzten Fall sollte die Fehlermeldung vom Bildschirm verschwinden nachdem der Benutzer auf den grauen Hintergrund drückt.

In der Abbildung 5.9 ist ein Ausschnitt aus der Umfrage mit Kommentaren zu sehen.

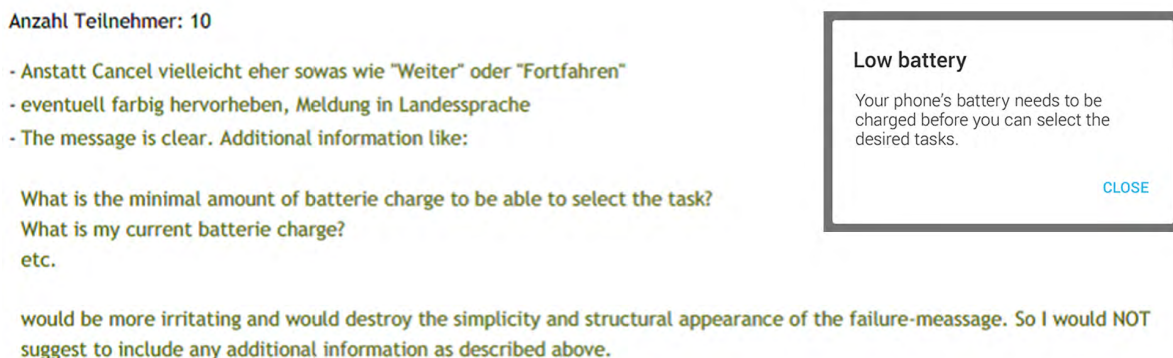


Abbildung 5.9.: Verbesserungsvorschläge zur "Low battery"-Fehlermeldung während der Aufgabenauswahl in der Worklist-View

Fehlermeldung "Low battery" während der Aufgabenausführung

Zu dieser "Low battery"-Fehlermeldung in der Task Execution Worklist-View (siehe Kapitel 4.2.1.1, Abb. 4.5 und Abb. 4.6) gab es ähnliche Verbesserungsvorschläge wie oben. Eine Stu-

dentin konnte die vorherige Meldung fließender lesen. Die anderen an der Umfrage beteiligten Personen rieten, eine Symbolik für den "Close"-Button oder eine bessere Erklärung des Problems einzuführen.

Der folgende Ausschnitt aus der Umfrage bildet die Antworten der Befragten ab.

Anzahl Teilnehmer: 7

- wie vor farbe, landessprache
- Symbolik für Close
- I think I was able to read the text of the previous error report more fluently.
- Schließt "close" die Fehlermeldung oder die App?
- Eine Begründung wäre nett
- Detailliertere Meldung
- Nichts auszusetzen. Vllt anstelle von close ein x oben rechts oder so ...

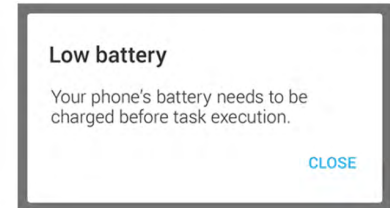


Abbildung 5.10.: Verbesserungsvorschläge zur "Low battery"-Fehlermeldung während der Aufgabenausführung in der Worklist-View

Fehlermeldung "Low battery" während der Aufgabendelegation

Die meisten Missverständnisse rief die im Kapitel 4.2.1.1 in der Abbildung 4.8 dargestellte Fehlermeldung zum Speichern der Zwischenergebnisse hervor. So empfahl ein Wirtschaftsphysiker eine andere Beschreibung des Problems auszuarbeiten und nannte die aktuelle Mitteilung als "ein netter Ratschlag zur Fortschrittsspeicherung" (siehe Abb. 5.11). Sein Tipp wäre, einen "Save"-Button anzubieten, der unter der Meldung über notwendige Akkuladung vor der Aufgabendelegation stehen könnte. Medieninformatiker fanden die Benachrichtigung ebenso "irgendwie total schlecht", "komisch" und "viel zu formal". Es wurde eine konkrete Lösung angeboten: "Speichere deine Liste bevor dein Akku leer ist".

Ferner tritt ein weiteres Problem in Erscheinung - eine Doppelbedeutung des Buttons "Close". Man sollte in der Fehlermeldung deutlicher darauf hinweisen, "dass mit close die liste verloren geht...denn wenn close wie bisher im kontext von schließen der meldung ist, dann hat man 1 benennung zu 2 kontexten". Die Bezeichnung des oben genannten Buttons war ebenfalls für einen Informatiker nicht aussagekräftig genug. Er fragte sich: "Was passiert bei close?"

Die Wirtschaftsmathematikerin korrigierte den Satzbau der Meldung auf folgende Weise: "Save the list of tasks, THAT should be delegated."

Anzahl Teilnehmer: 6

- Was passiert bei close?

- Isn't it rather:

"Save the list of tasks, THAT should be delegated.

- Andere Fehlermeldung zur Lösung des Problems (Low battery). Aktuelle Fehlermeldung ist ein netter Ratschlag zur Fortschrittsspeicherung, der vielleicht mit dem "save" Button unter der Meldung von Abbildung 4 stehen könnte.

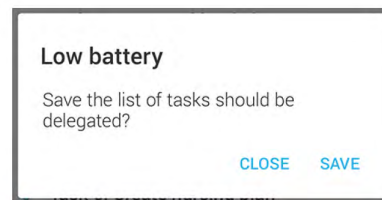


Abbildung 5.11.: Verbesserungsvorschläge zur "Low battery"-Fehlermeldung während der Aufgaben-delegation in der Worklist-View

5.3.2.2. Execution-View

Fehlermeldung "Low battery" während der Aufgabenausführung (Stufe 2)

Zu dem Fehler, der bei dem Erfassen der Benutzerdaten eintreffen könnte (siehe Kapitel 4.2.1.1, Abb. 4.10 und Abb. 4.12), hinterließen lediglich drei Studenten einen Kommentar.

Ein Informatiker interessierte sich dafür, ob die Daten nach dem Klicken auf den "Close"-Button temporär gespeichert werden und ob der Prozess sich später fortführen lässt. Weiterhin war die Fehlermeldung für einen Medieninformatiker "eindeutig zu lang". "Wenn der Akku fast leer ist kann der Nutzer auch mal in Panik geraten und dann aus versehen auf den falschen Button klicken", begründete er seine Meinung.

Die Wirtschaftsmathematikerin bezweifelte die Notwendigkeit der "Delegate"-Schaltfläche, falls ihr Smartphone dringend aufgeladen werden muss. Sie ging davon aus, dass nach dem Drücken dieses Buttons mehrere weiteren angeklickt werden müssen. Es war für sie schwer nachvollziehbar, warum es genug Zeit für die Delegation gibt, die Zeit aber nicht ausreicht, um das Formular vollständig auszufüllen. Denn es könnte, ihrer Vorstellung nach, sein, dass der Nutzer mit der Aufgabenausführung fast fertig war.

Die Abbildung 5.12 zeigt einige Anmerkungen zu der oben aufgeführten Fehlermeldung.

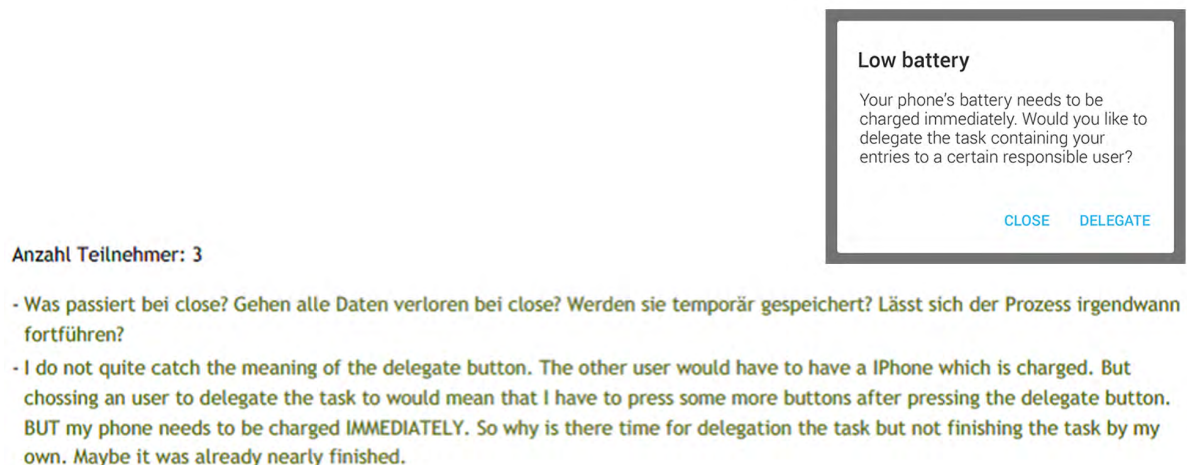


Abbildung 5.12.: Verbesserungsvorschläge zur "Low battery"-Fehlermeldung (2. Stufe) während der Aufgabenausführung in der Execution-View

Fehlermeldung "Critical low battery" während der Aufgabenausführung (Stufe 3)

Hier waren die Anmerkungen teilweise gegensätzlich. Während die Wirtschaftsmathematikerin dank der aktuellen Nachricht (siehe Kapitel 4.2.1.1, Abb. 4.10 und Abb. 4.13) eine Antwort auf ihre Frage zu der vorherigen Fehlermeldung bekam, empfand der Wirtschaftsphysiker die Mitteilung als "schwierig zu lesen". Außerdem wies er darauf hin, dass das Wort "Critical" "zu Panik und Nichtlesen der Fehlermeldung" führen könnte. Seiner Empfehlung zufolge, sollte es weglassen werden.

Ein Student der Medieninformatik bemerkte: "Trotz der Länge ist der Text gut verständlich". Im Gegenteil dazu schlug ein anderer Teilnehmer der Umfrage eine "Kürzung des Textes" vor.

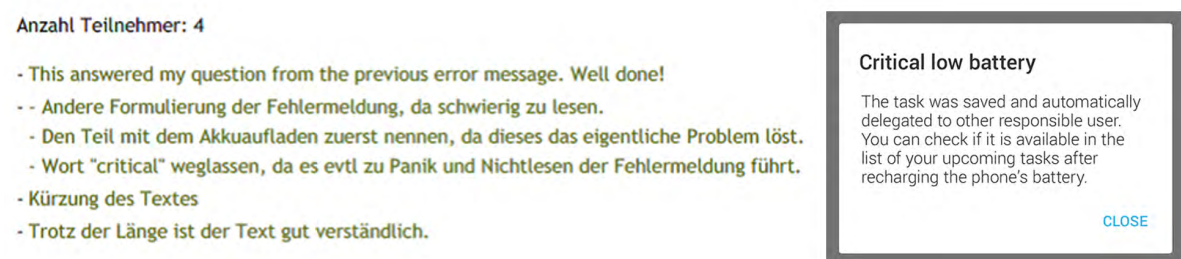


Abbildung 5.13.: Verbesserungsvorschläge zur "Critical low battery"-Fehlermeldung (3. Stufe) während der Aufgabenausführung in der Execution-View

Fehlermeldung "Low battery" während der Aufgabendelegation (Stufe 1) Außer einer sehr positiven Bewertung von einem Medieninformatiker gab es zu dieser "Low battery"-Fehlermeldung (siehe Kapitel 4.2.1.1, Abb. 4.14 und Abb. 4.15) nur noch einen Vorschlag von der Wirtschaftsmathematikerin, einen "Sofort delegieren"-Button einzuführen (siehe Abb. 5.14).

Anzahl Teilnehmer: 2

- Maybe change the choices to delegate now ... and ... close
- Sehr gut

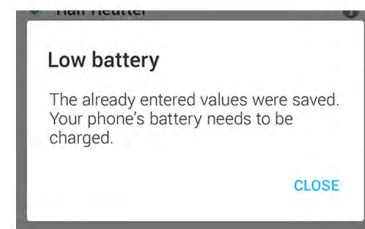


Abbildung 5.14.: Verbesserungsvorschläge zur "Low battery"-Fehlermeldung (1. Stufe) während der Aufgabendelegation in der Execution-View

Fehlermeldung "Critical low battery" während der Aufgabendelegation (Stufe 2)

Die Mitteilung über eine automatische Delegation an einen anderen verantwortlichen Benutzer (siehe Kapitel 4.2.1.1, Abb. 4.14 und Abb. 4.16) erzeugte weitere Kommentare. Der Wirtschaftsphysiker war der Ansicht, dass die Fehlermeldung nicht das Problem des kritischen Akkuzustands löst. Eine "Erwähnung des Akkuauf ladens" wäre für ihn sinnvoller. Des Weiteren hielt die Studierende der Wirtschaftsmathematik eine unerwünschte Delegation für möglich. Die Meinung teilte ein Medieninformatiker, denn er schlug eine Zustands zwischenspeicherung als Verbesserung vor.

Die Abbildung 5.15 stellt die Antworten der Befragten dar.

Anzahl Teilnehmer: 3

- Maybe I wanted to choose another user. I think automatical delegation could end up with unwanted delegation.
- Die Fehlermeldung löst nicht den kritischen Akkuzustand, sondern hilft mir nur bei der process App --> Erwähnung des Akkuauf ladens
- Speichern des Zwischenstandes ist besser

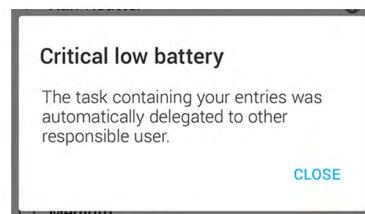


Abbildung 5.15.: Verbesserungsvorschläge zur "Critical low battery"-Fehlermeldung (2. Stufe) während der Aufgabendelegation in der Execution-View

6

Verbesserungspotential

In diesem Kapitel werden Verbesserungsvorschläge für die Textgestaltung sowie Benennung der Buttons vorgestellt. Diese können der weiteren Optimierung der Fehlerbehandlung dienen.

6.1. Verbesserungsvorschläge

Bei der Umfrage stellte sich heraus, dass sich manche Befragten mehr Details über den aufgetretenen Fehler wünschten (siehe Kapitel 5.3.2). Für den Anwender wäre es mit Sicherheit nützlich, wenn die Meldungen weitere Auskünfte geben würden, wie zum Beispiel über den geforderten Akkustand oder über die Nutzungsmöglichkeiten während des Akkuladens.

Die Design Guidelines aller drei Betriebssysteme raten jedoch von langen Meldungen ab. Ausführliche Texte erschweren das schnelle Lesen und beeinflussen die Nutzererfahrung negativ (siehe Kapitel 3). Es wird nachdrücklich darauf hingewiesen, die Benachrichtigungen prägnant zu halten. Der gleicher Meinung waren einige Umfrageteilnehmer, die bereits die kurzgefassten Warnungen reduzieren wollten (siehe Anhang A.3).

Es empfiehlt sich daher die bestehenden Fehlermeldungen eher zu verkürzen, als noch umfangreicher zu gestalten (siehe Kapitel 6.2).

Bezüglich der Buttons konnte ebenso Verbesserungspotential in der Umfrage aufgezeigt werden. Kommentare der Befragten deuteten darauf hin, dass der Button "Close" zur Verwirrung der User führen könnte. Es wurde nicht verstanden, ob diese Schaltfläche die Fehlermeldung oder die Anwendung schließt. Darüber hinaus schrieb ein Umfrageteilnehmer eine berechtigte Anmerkung über die Doppelbedeutung des "Close"-Buttons (siehe Seite 39).

Deshalb wäre es sinnvoll die Bezeichnung fallabhängig anzupassen. Es bietet sich an, den "Close"-Button durch den "OK"-Button zu ersetzen, wenn vom Nutzer eine einfache Akzeptanz erwartet wird (siehe iOS Empfehlungen auf der Seite 11 und Kommentare der Befragten im Anhang A.3). Für die weiteren "Close"-Buttons sollten präzisere Benennungen ausgesucht werden, die dahinter stehende Handlungen eindeutig definieren (siehe Kapitel 6.2).

Da die "Low battery"-Fehlermeldung während der Aufgabendelegation in der Execution-View starker Kritik ausgesetzt wurde, sollte diese umkonzipiert werden. Es wäre wichtig, die Problemlösung als Erstes zu erwähnen. Die Nicht-Speicherung der Delegationsliste könnte man mit dem "Reject"-Button bejahen (siehe Kapitel 6.2, Abb. 6.3 und Abb. 6.4).

6.2. Verbesserte Mockups

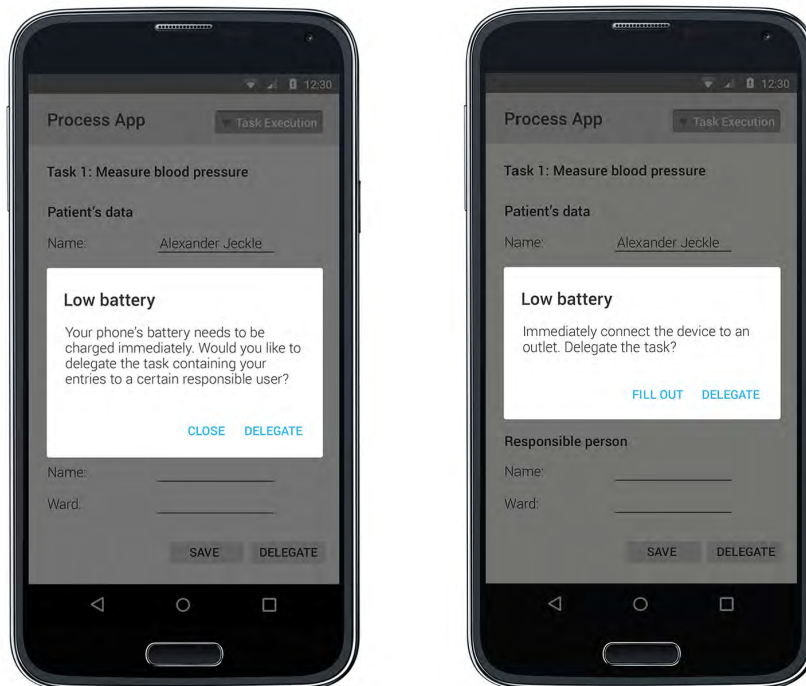


Abbildung 6.1.: Verbesserungsvorschlag zu der "Low battery"-Fehlermeldung während der Aufgaben-
ausführung in der Ausführungsansicht: Vorher (links) und nachher (rechts) - Android

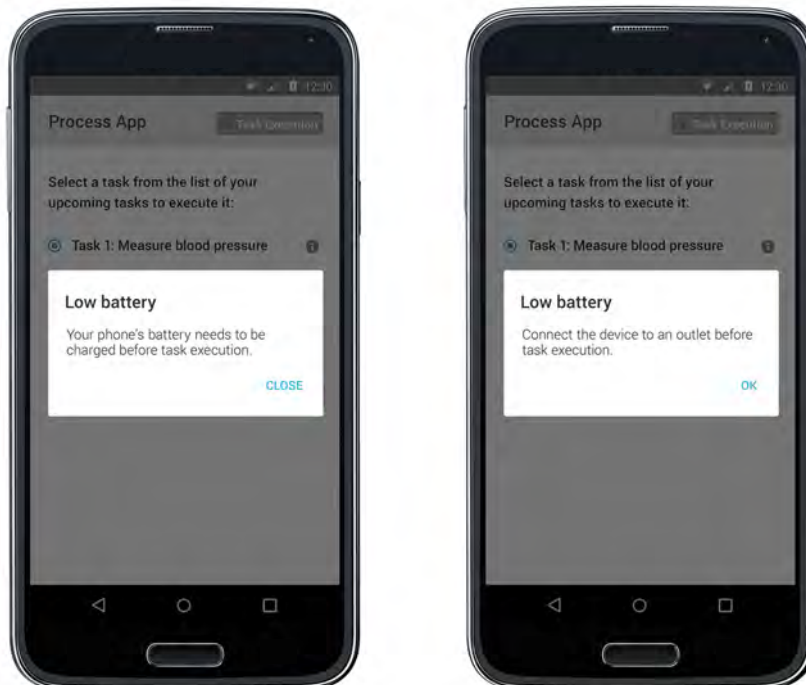


Abbildung 6.2.: Verbesserungsvorschlag zu der "Low battery"-Fehlermeldung während der Aufgaben-
ausführung in der Arbeitslistenansicht: Vorher (links) und nachher (rechts) - Android

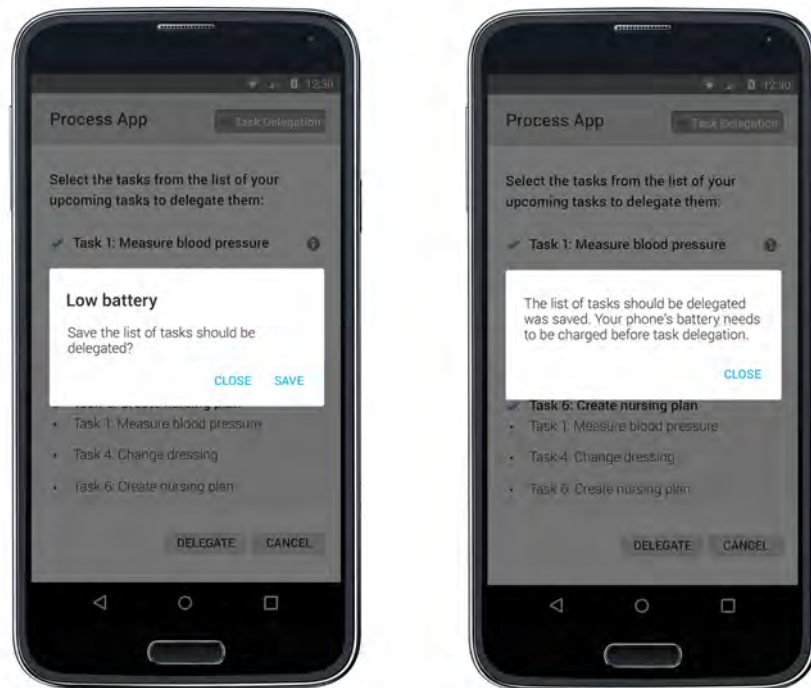


Abbildung 6.3.: Vorherige "Low battery"-Fehlermeldung während der Aufgabendelegation in der Arbeitslistenansicht - Android

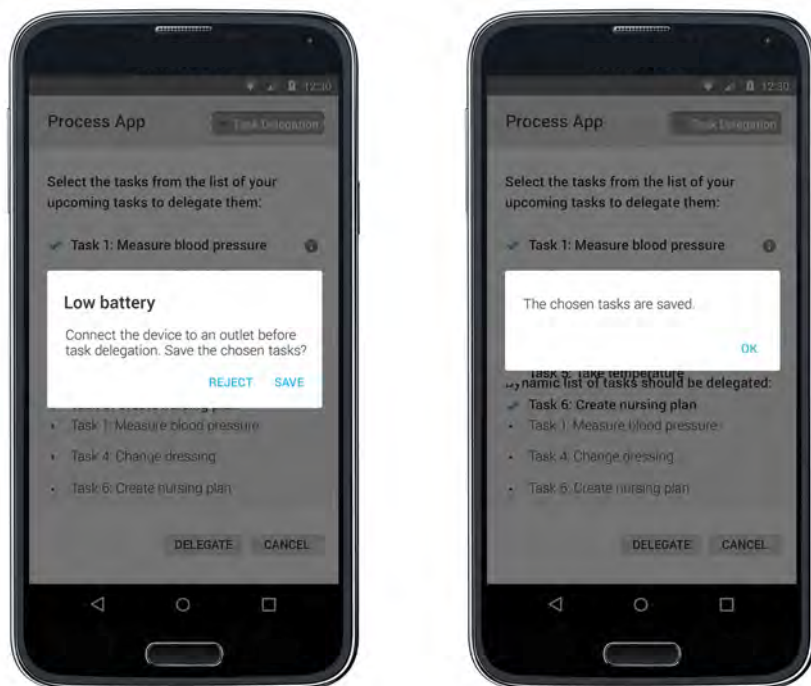


Abbildung 6.4.: Verbesserte "Low battery"-Fehlermeldung während der Aufgabendelegation in der Arbeitslistenansicht - Android

7

Fazit

Dieses Kapitel fasst die wichtigsten Aspekte der vorliegenden Bachelorarbeit zusammen. Die Fehlerbehandlung von mobilen Prozess-Aktivitäten liegt in Form von elektronischen Mockups vor. Sie sind unter Berücksichtigung der Design Guidelines der drei Betriebssysteme (Android 5.0, iOS 8 und Windows Phone 8) gestaltet worden.

Des Weiteren wurden die elektronischen Entwürfe auf Benutzerfreundlichkeit und Verständlichkeit in einer online Umfrage geprüft. Die Befragung brachte wertvolle Ergebnisse für die Weiterentwicklung des Usability-Konzepts. Die Teilnehmer bewerteten die Optik und Bedienung der Fehlermeldungen weitgehend positiv. Noch wichtiger: Die vor dem Arbeitsbeginn zur Umsetzung vorgegebenen Delegations- und Backup-Konzepte wurden von den Umfrageteilnehmern verstanden.

Nichts desto trotz gab es einige Kritikpunkte, die zu weiteren Überlegungen bezüglich der Konzipierung der Meldungen führten. Als Ergebnis ist ein extra Kapitel namens Verbesserungspotential (siehe Seiten 43 bis 45) entstanden, das einige Optimierungsvorschläge aufführt. Die Ideen sollten in Zukunft berücksichtigt und falls notwendig auf die restlichen Fehlermeldungen übertragen werden.

Es empfiehlt sich, weitere Untersuchungen in diesem Themengebiet durchzuführen. Insbesondere wären zusätzliche Umfragen von großer Bedeutung, um die Meinung der User zu den im Anhang A.2 befindlichen Mockups herauszufinden.

Literaturverzeichnis

- [CNB⁺13] CROMBACH, A. ; NANDI, C. ; BAMBONYE, M. ; LIEBRECHT, M. ; PRYSS, R. ; REICHERT, M. ; ELBERT, T. ; WEIERSTALL, R.: Screening for mental disorders in post-conflict regions using computer apps - a feasibility study from Burundi. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, 2013, S. 70–70
- [Cora] CORPORATION, Microsoft: *Design resources for Windows Phone*. [https://msdn.microsoft.com/library/windows/apps/ff637515\(v=vs.105\).aspx](https://msdn.microsoft.com/library/windows/apps/ff637515(v=vs.105).aspx), . – [letzter Abruf: 20.09.2015]
- [Corb] CORPORATION, Microsoft: *Error Messages*. [https://msdn.microsoft.com/en-us/library/windows/desktop/dn742471\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dn742471(v=vs.85).aspx), . – [letzter Abruf: 20.09.2015]
- [Corc] CORPORATION, Microsoft: *Guidelines for message dialogs*. <https://msdn.microsoft.com/en-us/library/windows/apps/hh738363.aspx>, . – [letzter Abruf: 20.09.2015]
- [Cord] CORPORATION, Microsoft: *Guidelines for message dialogs*. <https://msdn.microsoft.com/en-us/library/windows/apps/hh738363.aspx>, . – [letzter Abruf: 20.09.2015]
- [Core] CORPORATION, Microsoft: *Richtlinien für Dialogfeld-Steuerelemente*. <https://msdn.microsoft.com/de-de/library/windows/apps/Dn997764.aspx>, . – [letzter Abruf: 20.09.2015]
- [Corf] CORPORATION, Microsoft: *Richtlinien für UWP-Apps*. <https://msdn.microsoft.com/de-de/library/windows/apps/hh465424.aspx>, . – [letzter Abruf: 20.09.2015]
- [Corg] CORPORATION, Microsoft: *Richtlinien zum Entwerfen barrierefreier Apps*. <https://msdn.microsoft.com/de-de/library/windows/apps/Hh700407.aspx>, . – [letzter Abruf: 20.09.2015]
- [Gmb] GMBH enuvo: *Umfrage Online*. <https://www.umfrageonline.com>, . – [letzter Abruf: 20.09.2015]
- [GPSR13] GEIGER, P. ; PRYSS, R. ; SCHICKLER, M. ; REICHERT, M.: Engineering an Advanced Location-Based Augmented Reality Engine for Smart Mobile Devices. Ulm : University of Ulm, October 2013 (UIB-2013-09). – Technical Report
- [GSP⁺14] GEIGER, P. ; SCHICKLER, M. ; PRYSS, R. ; SCHOBEL, J. ; REICHERT, M.: Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned. In: *10th Int'l Conference on Web Information Systems and Technologies*, 2014, S. 383–394
- [Inca] INC, Apple: *Accessibility Programming Guide for iOS*. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/iPhoneAccessibility/Introduction/Introduction.html>, . – [letzter Abruf: 20.09.2015]

- [Incb] INC, Apple: *Alert Views*. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UIAlertView.html>, . – [letzter Abruf: 20.09.2015]
- [Incc] INC, Apple: *Temporary Views*. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Alerts.html>, . – [letzter Abruf: 20.09.2015]
- [Incd] INC, Google: *Accessibility*. <https://www.google.com/design/spec/usability/accessibility.html>, . – [letzter Abruf: 20.09.2015]
- [Ince] INC, Google: *Dialogs*. <https://www.google.com/design/spec/components/dialogs.html>, . – [letzter Abruf: 20.09.2015]
- [Incf] INC, Google: *Material Design Icons 2.0*. <https://github.com/google/material-design-icons/releases>, . – [letzter Abruf: 20.09.2015]
- [IRLP⁺13] ISELE, D. ; RUF-LEUSCHNER, M. ; PRYSS, R. ; SCHAUER, M. ; REICHERT, M. ; SCHOBEL, J. ; SCHINDLER, A. ; ELBERT, T.: Detecting adverse childhood experiences with a little help from tablet computers. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, 2013, S. 69–70
- [Kue] KUEHNEL, Andreas: *Visual C# 2008*. http://openbook.rheinwerk-verlag.de/visual_csharp/visual_csharp_13_009.htm, . – [letzter Abruf: 20.09.2015]
- [LLC] LLC, Pixeden: *Samsung Galaxy S5 Psd Mockup*. <http://www.pixeden.com/psd-mock-up-templates/samsung-galaxy-s5-psd-mockup>, . – [letzter Abruf: 20.09.2015]
- [Ltd] LTD, Adobe Systems Software I.: *Adobe-Downloads*. <http://www.adobe.com/de/downloads.html>, . – [letzter Abruf: 20.09.2015]
- [PLRH12] PRYSS, R. ; LANGER, D. ; REICHERT, M. ; HALLERBACH, A.: Mobile Task Management for Medical Ward Rounds - The MEDo Approach. In: *1st Int'l Workshop on Adaptive Case Management*, Springer, September 2012 (LNBIP 132), S. 43–54
- [PMLR15] PRYSS, R. ; MUNDBROD, N. ; LANGER, D. ; REICHERT, M.: Supporting medical ward rounds through mobile task and process management. In: *Information Systems and e-Business Management* 13 (2015), February, Nr. 1, S. 107–146
- [PMR13] PRYSS, R. ; MUSIOL, S. ; REICHERT, M.: Collaboration Support Through Mobile Processes and Entailment Constraints. In: *9th IEEE Int'l Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'13)*, IEEE Computer Society Press, October 2013
- [PMR14] PRYSS, R. ; MUSIOL, S. ; REICHERT, M.: Integrating Mobile Tasks with Business Processes: A Self-Healing Approach. In: *Handbook of Research on Architectural Trends in Service-Driven Computing*. 2014, S. 103–135
- [PTKR10] PRYSS, R. ; TIEDEKEN, J. ; KREHER, U. ; REICHERT, M.: Towards Flexible Process Support on Mobile Devices. In: *Proc. CAiSE'10 Forum - Information Systems Evolution*, Springer, 2010 (LNBIP 72), S. 150–165

- [PTR10] PRYSS, R. ; TIEDEKEN, J. ; REICHERT, M.: Managing Processes on Mobile Devices: The MARPLE Approach. In: *CAiSE'10 Demos*, 2010
- [RLPL⁺13] RUF-LEUSCHNER, M. ; PRYSS, R. ; LIEBRECHT, M. ; SCHOBEL, J. ; SPYRIDOU, A. ; REICHERT, M. ; SCHAUER, M.: Preventing further trauma: KINDEX mum screen - assessing and reacting towards psychosocial risk factors in pregnant women with the help of smartphone technologies. In: *XIII Congress of European Society of Traumatic Stress Studies (ESTSS) Conference*, 2013, S. 70–70
- [RPR11] ROBECKE, A. ; PRYSS, R. ; REICHERT, M.: DBIScholar: An iPhone Application for Performing Citation Analyses. In: *CAiSE Forum-2011*, CEUR Workshop Proceedings, June 2011 (Proceedings of the CAiSE'11 Forum at the 23rd International Conference on Advanced Information Systems Engineering Vol-73)
- [SPR15] SCHOBEL, J. ; PRYSS, R. ; REICHERT, M.: Using Smart Mobile Devices for Collecting Structured Data in Clinical Trials: Results From a Large-Scale Case Study. In: *28th IEEE International Symposium on Computer-Based Medical Systems (CBMS 2015)*, IEEE Computer Society Press, June 2015, S. 13–18
- [SPSR15] SCHICKLER, M. ; PRYSS, R. ; SCHOBEL, J. ; REICHERT, M.: An Engine Enabling Location-based Mobile Augmented Reality Applications. In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014)*. Springer, 2015 (LNBIP)
- [SSP⁺14a] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; MAIER, F. ; REICHERT, M.: Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned. In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014)*, 2014, S. 371–382
- [SSP⁺14b] SCHOBEL, J. ; SCHICKLER, M. ; PRYSS, R. ; MAIER, F. ; REICHERT, M.: Towards Process-Driven Mobile Data Collection Applications: Requirements, Challenges, Lessons Learned. In: *10th Int'l Conference on Web Information Systems and Technologies (WEBIST 2014)*, 2014, S. 371–382